MULTIPROCESSING ... OPEN SOURCE DEVELOPMENT PROCESS ... OCP-IP and OSCI

Embedded COMPUTING DESIGN

www.embedded-computing.com

JANUARY 2005 Vol. 3 No. 1

SPECIAL FEATURES:

Eclipse: An open and extensible IDE

GUEST FEATURE:

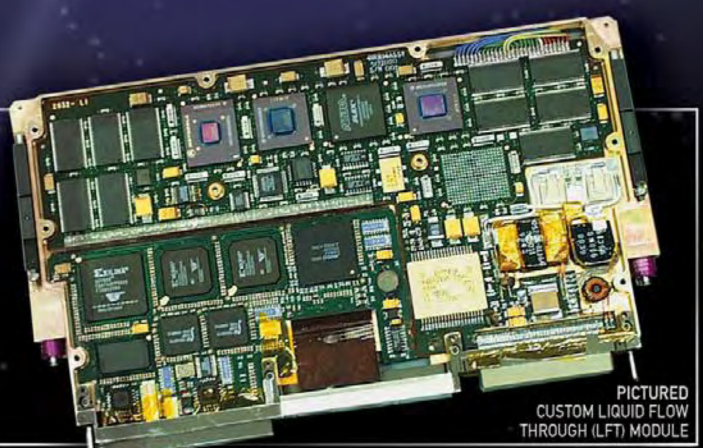
FPGAs are poised for a change

TECHNOLOGY FEATURE:

Military and commercial SBC cooling

APPLICATION FEATURE:

Embedded system behavior: RTOS vs. GPOS



PICTURED
CUSTOM LIQUID FLOW
THROUGH (LFT) MODULE

How to turn a flood of video into a stream.

TSUNAMI PMC

- Numerous interface options on mezzanine
- Format, filter and combine raw signals
- Compress with MPEG, JPEG integrated IP
- Powerful FPGA-based processing
- Conduction cooled or commercial grades
- LVDS or RS170 I/O standard

SBS Knows FPGA Computing. And for many video applications, nothing outperforms programmable logic.

Video-based applications like surveillance, unmanned flight, radar, sonar, inspection and medical imaging demand high-speed video processing in real time, and our new FPGA-based PMC is designed for just that sort of work.

It features a powerful FPGA processor, supported by four high-speed, high-capacity memory banks with an aggregate memory bandwidth of over 1.8 Gbytes/second. To put that in practical terms, this board can perform multiple streams of MPEG4 compression at full video rates direct from your camera to your network.

SBS has made the integration process painless. Our bundled products deliver out-of-the-box RS170 and LVDS video capture and compression solutions. Need more? Easy-to-use development kits include APIs, drivers, memory controllers, source code and application examples to deliver quick results. Multiple operating systems are supported, and we also offer the SBS FPGA Accelerator Program which further simplifies the integration process. So give us a call, and let us help you tame your video.

SBS knows. To find out more, visit us at **www.sbs.com** or call us at **800.SBS.1553**

SBS Technologies®

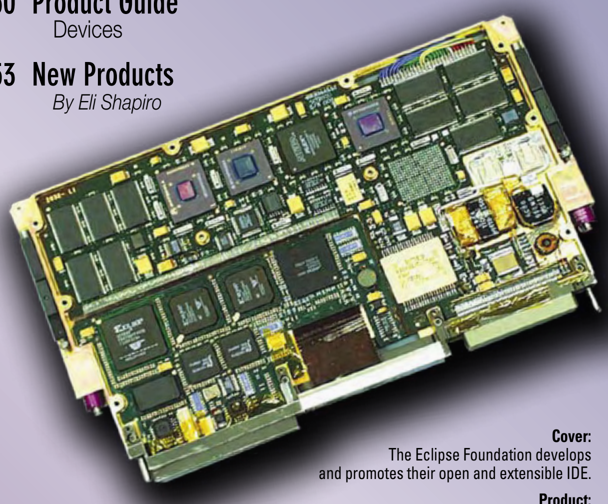
Embedded COMPUTING DESIGN

WWW.EMBEDDED-COMPUTING.COM

Volume 3 • Number 1
JANUARY 2005

CONTENTS

COLUMNS



Cover:
The Eclipse Foundation develops and promotes their open and extensible IDE.
Product:
Cooling example. A custom Liquid Flow Through (LFT) module.

EVENTS

Bus & Board 2005 Conference
January 17-18 • Long Beach, CA
www.busandboard.com

Measurement Science Conference
January 17-21 • Anaheim, CA
www.msc-conf.com

Embedded Computing Design published by: OpenSystems Publishing™

FEATURES

SPECIAL: Eclipse IDE Platform

APPLICATION: RTOS Trends

TECHNOLOGY: Cooling Trends

GUEST: FPGA Trends

E-LETTERS ONLINE

January: Next-generation embedded Linux development tools for high reliability and mission-critical applications
By William von Hagen, TimeSys Corporation
www.embedded-computing.com/eletter

FREE SUBSCRIPTIONS

Subscribe to the magazine or E-letter at:
www.opensystems-publishing.com/subscriptions

WWW
embedded-computing.com

Headless, But Not Mindless.

Add Micro/sys brainpower to your embedded application.
There's a lot riding on that new OEM design: sophisticated application code, tricky I/O interfaces, and system integration issues. You don't need a headache with CPU design issues, too. You just want a reliable CPU that will plug into your application, interface with your system I/O, and run. Headless might be the right way to go, but the price has got to be right.

We have just the processing brainpower you need!
For example, the new Micro/sys SBC1586 is a headless PC/104 size Pentium® workhorse. It includes 10/100BASE-T Ethernet, USB, four COM ports and a removable CompactFlash® for data logging or OS. This embedded PC supports Linux, DOS, Windows® NT or CE, and VxWorks®. Starting under $500.

For a less scary ride on your development adventure, choose from a wide selection of X86 through Pentium CPUs, headless or fully configured with A/D, D/A, and other PC I/O. Call today and let the minds of our engineering and customer service teams help you rein-in those unbridled development costs by offering a compact and cost-effective solution.

MICRO/SYS

3730 Park Place, Montrose, CA 91020
Voice (818) 244-4600 Fax (818) 244-4246
info@embeddedsys.com

www.embeddedsys.com

For more fully-featured Micro/sys embedded PCs in various sizes, options and interfaces, take a look at the boards below and check our Web site for complete details:

SBC1586 (right)
Headless Low-Power Pentium PC/104 board with CompactFlash, 4 COM ports, USB and Ethernet. Starting under $500.

SBC1495 (middle)
PC/104 OEM 486/586 with VGA, COM, Ethernet, and CompactFlash with DOS, Windows, Linux OS.

SBC2590 (left)
EBX Pentium Controller with VGA, COM, Ethernet, A/D, D/A and your choice of Operating System.

Embedded COMPUTING DESIGN

WWW.EMBEDDED-COMPUTING.COM

An OpenSystems Publication

ISSN: Print 1542-6408, Online 1542-6459

Embedded Computing Design is published quarterly by OpenSystems Publishing LLC., 30233 Jefferson Ave., St. Clair Shores, MI 48082.

Subscriptions are free, upon request in writing, to persons dealing with or considering *Embedded Computing Design.* For others inside the U.S. and Canada, subscriptions are $24/year. For 1st class delivery outside the U.S. and Canada, subscriptions are $50/year (advance payment in U.S. funds required).

Canada: Publication agreement number 40048627
Return address: WDS, Station A, PO Box 54, Windsor, ON N9A 615

POSTMASTER: Send address changes to

Embedded Computing Design
13253 La Montana, Suite 207 • Fountain Hills, AZ 85268

OpenSystems Publishing™

SALES OFFICE 586-415-6500

VP OF MARKETING & SALES
Patrick Hopper
phopper@opensystems-publishing.com

SENIOR ACCOUNT MANAGER
Dennis Doyle
ddoyle@opensystems-publishing.com

PRINT/ONLINE MARKETING SPECIALIST
Christine Long
clong@opensystems-publishing.com

MARKETING REPRESENTATIVE
Andrea Stabile
astabile@opensystems-publishing.com

ACCOUNT MANAGERS
Doug Cordier
dcordier@opensystems-publishing.com

Tom Varcie
tvarcie@opensystems-publishing.com

EUROPEAN REPRESENTATIVE
Stefan Baginski
sbaginski@opensystems-publishing.com
+41(0)62 758-3222

FOR REPRINTS:
Call the Sales Office

ADVERTISER INDEX

FREE INFORMATION

PUBLISHERS
John Black, Michael Hopper, Wayne Kristoff

ADVERTISING/BUSINESS OFFICE
30233 Jefferson Avenue
St. Clair Shores, MI 48082
Tel: 586-415-6500 • Fax: 586-415-4882

EDITORIAL/PRODUCTION OFFICE
13253 La Montana, Suite 207
Fountain Hills, Arizona 85268
Tel: 480-967-5581 • Fax: 480-837-6466

SENIOR TECHNICAL EDITOR
Mark David Barrera
mbarrera@opensystems-publishing.com

TECHNICAL EDITOR
Chad Lumsden
clumsden@opensystems-publishing.com

NEWS EDITOR
bicrutcher@opensystems-publishing.com

NEW PRODUCTS EDITOR
Eli Shapiro
newproducts@opensystems-publishing.com

ASSOCIATE EDITOR
Anne Fisher

VICE PRESIDENT OF EDITORIAL
Rosemary Kristoff
rkristoff@opensystems-publishing.com

MANAGING EDITOR
Bonnie Crutcher

CONTRIBUTING WRITERS
Markus Levy
Bill Weinberg

SENIOR DESIGNER
Joann Toth

SENIOR WEB DESIGNER
Konrad Witte

GRAPHIC SPECIALIST
David Diomede

BUSINESS MANAGER
Karen Layman

CIRCULATION/OFFICE MANAGER
Phyllis Thompson
subscriptions@opensystems-publishing.com

Current Development Trends

Mark David Barrera

Welcome to the January trends issue of *Embedded Computing Design.* Longtime readers will note that ECD is now a bi-monthly publication, offering six issues a calendar year. This welcome development means that you will be reading one additional *Editor's Foreword* this year. In this issue, the feature articles illustrate the wide range of current embedded development trends.

Eclipse IDE Platform

- *Eclipse: The development system that crosses RTOS boundaries*, by Robert Day of Accelerated Technology, describes the Eclipse Platform. Eclipse is an open platform for tool integration built by an open community of tool providers. To quote the Eclipse website: "The Eclipse Platform is an open IDE for anything, and for nothing in particular."

- *Adding value with the Eclipse framework*, by Michael McCullough of MCC Systems, describes the Isolationist, Proprietary, or Truly Open approaches that have been taken by vendors when incorporating Eclipse into deliverable products.

RTOS trends

- *RTOS versus GPOS: What is best for embedded development?*, by Paul N. Leroux of QNX Software Systems, questions if most embedded projects still need an RTOS. The article describes the impact of the recent introduction of real-time patches for Linux, Windows, and other General Purpose Operating Systems (GPOSs).

Cooling trends

- *Taking the heat: Strategies for cooling SBCs in commercial and military environments*, by Ivan Straznicky and Phuc Nguyen of Curtiss-Wright Controls Embedded Computing, discusses current strategies for successful cooling of Single Board Computers (SBCs). The choice of strategy is dependent upon whether the system is to be deployed in an air-cooled commercial environment (industrial or laboratory class), or a conduction-cooled rugged environment (defense and aerospace).

FPGA trends

- *FPGAs are poised for a change*, by Steve Mensor of Altera, discusses how the market shift to programmable logic will only accelerate as more advanced process technologies become available.

As always, I encourage your comments and suggestions concerning this and future issues. Please do not hesitate to send in an article abstract for any complex embedded systems subject.

M. D. Barrera

Mark David Barrera
Sr. Electrical Engineer
Sr. Technical Editor
OpenSystems Publishing

mbarrera@opensystems-publishing.com

LOOKING FORWARD

The March edition is our annual embedded products *Buyer's Guide*. Please see the editorial calendar (embedded-computing.com) for product or article submission information.

By Markus Levy

Multiprocessing becomes mainstream

Multiprocessing systems have been around since the beginning of time, or so it seems. Led by system and processor companies such as Intel, SGI, and Sun, multiprocessing has always had a home in workstations and servers. Recently, AMD and Intel have started down the multiprocessing path for PCs as well with innovations that make it more practical to put more than one PC processor core on a single die.

In the embedded market, the multicore approach has long been a viable technology. For example, Freescale and Texas Instruments have their dual-core MXC and OMAP architectures, respectively, that combine an ARM core and a DSP core. PMC-Sierra has its dual-core MIPS64-compatible RM9000 and recently announced the 1.8 GHz RM11200. Freescale has recently jumped on board with its dual-core PowerPC 8641D that integrates two e600 CPU cores. The list of multiprocessor devices goes on and on.

The need for multiprocessing

The best-known method of multiprocessing is referred to as Symmetric Multiprocessing (SMP) or *homogenous* multiprocessing. This method is represented by systems with two or more identical processors, or systems with a processor with two or more identical cores.

The obvious benefit of this method is a theoretical doubling of performance (or multiplication by a factor of n, where n is the number of processors). This is useful for increasing scalability, improving system density, boosting processing power without the incremental costs of support chips, and providing true concurrency. SMP has recently seen an increase in popularity for PC applications, as AMD and Intel have hit the operating frequency barrier due to power consumption issues.

Multiprocessing can also be implemented using an asymmetrical or *heterogeneous* method. Asymmetrical Multiprocessing (AMP) allows the system designer to use the processor best suited for a specific task or group of tasks. For mobile phone applications, the previously mentioned MXC and OMAP architectures use the ARM processor for running the application code and user interface, and the DSP processor for modem functions and accelerating multimedia algorithms such as MPEG-4 decode.

Hardware-based multithreading is a virtual method of multiprocessing that takes advantage of a single processor's built-in hardware support to simultaneously run multiple concurrent tasks and take advantage of idle cycles. Useful for implementing fine-grained multithreading, this method of multiprocessing is a good solution for the mismatch between processor speed and memory bandwidth. The basic premise behind multithreading is to minimize idle CPU cycles by executing several instruction streams simultaneously. When a thread encounters a cache miss, subsequent threads are activated to avoid any stall cycles.

Benefits for a wide range of applications

Theoretically, the hardware realization of any of these multiprocessing methods is relatively straightforward, but the true art of the deal is in making multiprocessing transparent to the system designer. In other words, the ultimate goal is to allow system designers to implement multiprocessing applications with no extra effort (or at least a minimal amount of effort). Hence, the partitioning burden is placed on the operating system, compiler, and other software tool vendors.

A big challenge for any of these tool vendors relates to the wide variety of applications where multiprocessing techniques can be applied. For example, in the consumer market, multiprocessing can be applied to the set-top box, telematics, smart phones, and gaming platforms. In the networking market, multiprocessing is useful for symmetrical packet processing, TCP termination offload, security processing, and Ethernet drivers (MACs).

Obviously, this wide variety of applications implies a wide variety of hardware and software multiprocessing techniques. The wide variety also implies a huge challenge in deriving industry-standard benchmarks to measure and compare the capabilities of the different processor and system solutions.

A simple (albeit ineffective) solution to benchmarking would be to measure the total throughput available from running multiple instances of either the same or different applications, while eliminating any inter-application dependencies. However, to create realistic usage scenarios of running multiple independent applications, any benchmark should run different applications to stress the platform's ability to support the multiple cache contexts

ARE YOU THINKING DIAMOND YET?

COMPACT EMBEDDED CPU WITH DATA ACQUISITION

NEW! ATHENA

The Athena extended temperature embedded CPU combines high performance, high integration, and rugged design into a small size (4.175" x 4.475"). Athena integrates the low-power Pentium-III class VIA Eden processor (400-660MHz) with a full range of standard I/O features as well as data acquisition. The result is a small, low-cost, yet extremely versatile embedded CPU that fits in tight spaces and survives harsh environments. With 128MB on-board memory, LCD+CRT video, AC97 audio, 4 USB ports, 4 serial ports, and PC/104 expansion, Athena is an all-in-one, complete solution.

ATHENA SYSTEM FEATURES

- VIA Eden Pentium-III class processor, 400-660MHz
- Integrated video, audio, Ethernet, and data acquisition
- 128MB memory soldered on board
- 4 RS-232 ports, 4 USB ports, PS/2 keyboard & mouse, Parallel and IDE port
- PC/104 expansion bus
- Operating temperature -40 to +85ºC
- Power dissipation: 9.5 W / fanless (400 MHz); 11 W / fan (660 MHz)
- Runs Linux, Windows, QNX, DOS, and VxWorks
- Mechanically and electrically compatible upgrade for Prometheus CPU
- Compatible with Pandora rugged system enclosure

DATA ACQUISITION FEATURES

- 16 analog inputs, 16-bit A/D, 100 KHz max sampling rate
- Programmable input ranges and A/D FIFO
- 4 analog outputs, 12-bit D/A
- 24 bi-directional digital I/O lines
- 2 counter/timers for A/D sampling and user operations
- Supported by Universal Driver software for Linux, Windows, QNX, and DOS

PC/104 ANALOG I/O MODULES

- 16-32 analog inputs
- 12-bit and 16-bit A/D
- 2-16 analog ouputs
- 12-bit and 16-bit D/A
- Autocalibration
- -40 to +85°C operation

PC/104 COMMUNICATION MODULES

EMM-OPTO NEW!

- 4 optically isolated serial ports
- RS-232 / RS-422 / RS-485 protocols
- 460.8kbaud maximum rate
- 24 programmable digital I/O
- -40 to +85°C operation

MERCURY NEW!

- 2 PCI-based 10/100Mbps Ethernet ports
- RJ-45 and locking pin headers for each port
- 24 programmable digital I/O
- -40 to +85°C operation

PC/104 POWER SUPPLIES

- 25-50 watts output power
- ±5V, ±12V outputs
- 8-30VDC inut range
- Battery charger available
- RS-232 / RS-422 / RS-485 on board!
- -40 to +85°C operation

THE DIAMOND SYSTEMS ADVANTAGE

We've enjoyed 15 years of uninterrupted growth helping customers around the world meet their embedded computing and analog measurement needs. Diamond Systems' boards are:

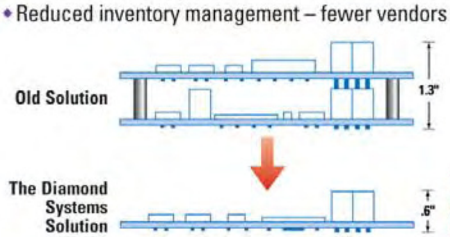
RUGGED

Our embedded computer boards are designed and tested for reliable operation in harsh environments. That's why we've been designed into military, space and vehicle applications around the world. Features like soldered on memory, jumperless configuration, and -40 to +85° operation mean your mission will be a success.

COMPACT

Diamond Systems reduces the size of your embedded system with our 2-in-1 and 3-in-1 boards. These boards prove important benefits:

- Reduced system size and weight
- Reduced assembly time – fewer boards to install and set up
- Reduced inventory management – fewer vendors and components



The 2-in-1 concept from Diamond Systems

ACCURATE

Autocalibration delivers maximum accuracy for your analog measurements over time and temperature. Our unique multi-range autocalibration technology calibrates each A/D input range independently to maintain accuracy when switching between ranges.

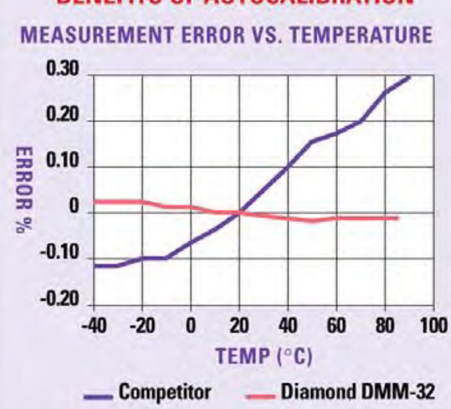
The diagram at right shows the actual performance of our autocalibrating A/D board vs. a competitor's manually-calibrated board. *Our board reduced temperature-based errors by a factor of 10!*

What's more, with autocalibration you can recalibrate both A/D and D/A circuits in just seconds under software control. No test equipment or handling is required. This means less downtime and lower maintenance costs.

DIAMOND SYSTEMS EMBEDDED BOARDS WITH AUTOCALIBRATION:

- Hercules EBX CPU
- DMM-48-AT A/D board
- DMM-AT A/D board
- DMM-16-AT A/D board
- DMM-32-AT A/D board

BENEFITS OF AUTOCALIBRATION



DIAMOND SYSTEMS CORPORATION
8430-D CENTRAL AVENUE
NEWARK, CA 94560
TEL: (510) 456-7800 www.diamondsystems.com

THINKING PC/104? THINK DIAMOND.

associated with multiple applications, as opposed to highlighting the ability to cache a single application across multiple processors.

A good multiprocessing benchmark could exploit a single application that has multiple task activities that are spread across the available processing resources. These applications can be identical, such as in networking, where the applications support multiple identical streams.

Alternatively, in a consumer-level device, the application could be processing multiple different streams (encode/decode for both audio/video streams), while processing network packets (TCP/IP), and controlling a user interface. These models of multiprocessing place very different load characteristics on the hardware that must maintain a consistent memory image of that application across the multiple processors.

Another good multiprocessing benchmark can be derived by using a single task that can be parallelized to be scalable across multiple instruction contexts. This type of benchmark must stress the system in terms of fine-grained synchronization access to shared resources.

Multiprocessing is a hot topic and represents a significant growth area for the embedded industry. EEMBC has embarked on a mission to develop industry-standard benchmarks that address the various methods of multiprocessing for the embedded market. Deviating from the consortium's standard procedure, it will be necessary to run these benchmarks on top of a common operating system API. Similar to the consortium's current mode of operation, these new benchmarks will follow an application-centric approach, although the choice of specific applications has not yet been determined. Stay tuned!

Markus Levy is founder and President of EEMBC. He is also Technical Editorial Director and Analyst at ConVergence Promotions. Mr. Levy received several patents while at Intel for flash memory architecture and for flash memory drives.

EEMBC – the Embedded Microprocessor Benchmark Consortium – was formed in 1997 to develop meaningful performance benchmarks for embedded system hardware and software. Contact the EEMBC directly for membership and certification information.

EEMBC

2222 Francisco Drive
Suite 510-203
El Dorado Hills, CA 95762
Tel: 530-672-9113
Fax: 530-672-9103
E-mail: markus@eembc.org
Website: www.eembc.org

The Open Source Development Process

By Bill Weinberg

As Linux and Open Source software becomes ever more popular in embedded applications, developers and management at device OEMs strive to understand the Open Source community, its practices, and their own companies' roles in Open Source. This column is intended to provide an overview of the processes and practices that drive Open Source development, both as a primer for the justification and use of Open Source Software (OSS) in embedded projects, and as an invitation to developers of all types to participate in those processes.

A disciplined process

Your company probably already uses an application or tool that was developed using Open Source methodologies. For example, your corporate website may use Apache with PHP or Perl, some portion of your corporate data may be warehoused with PostgreSQL, and you may now use or have probably used the Netscape browser. It is also extremely likely that your embedded projects are built with a member of the GNU Compiler Collection (GCC), and somewhere in your company you use Linux for enterprise or even embedded applications.

Open Source is not a free-for-all. Source code, while flowing freely from developers to distributions to end-users (and back) does not make its way helter-skelter into projects and from there into deployed devices. The Open Source development process is probably more disciplined than many proprietary software development processes. It has to be as the Open Source development process must be able to integrate, test, and quality-assure contributions from thousands of developers from around the globe.

The Linux kernel development process (Figure 1) in particular has been described as a *benevolent dictatorship*. Patches and enhancement are submitted for all parts of the Linux kernel by developers worldwide, thereby creating a borderless community. Vetting and integrating this diverse set of contributions falls to the 80 or so subsystem maintainers who focus on and contribute to particular aspects of the kernel such as file systems, memory management, and the system call interface.

LINUX KERNEL DEVELOPMENT PROCESS

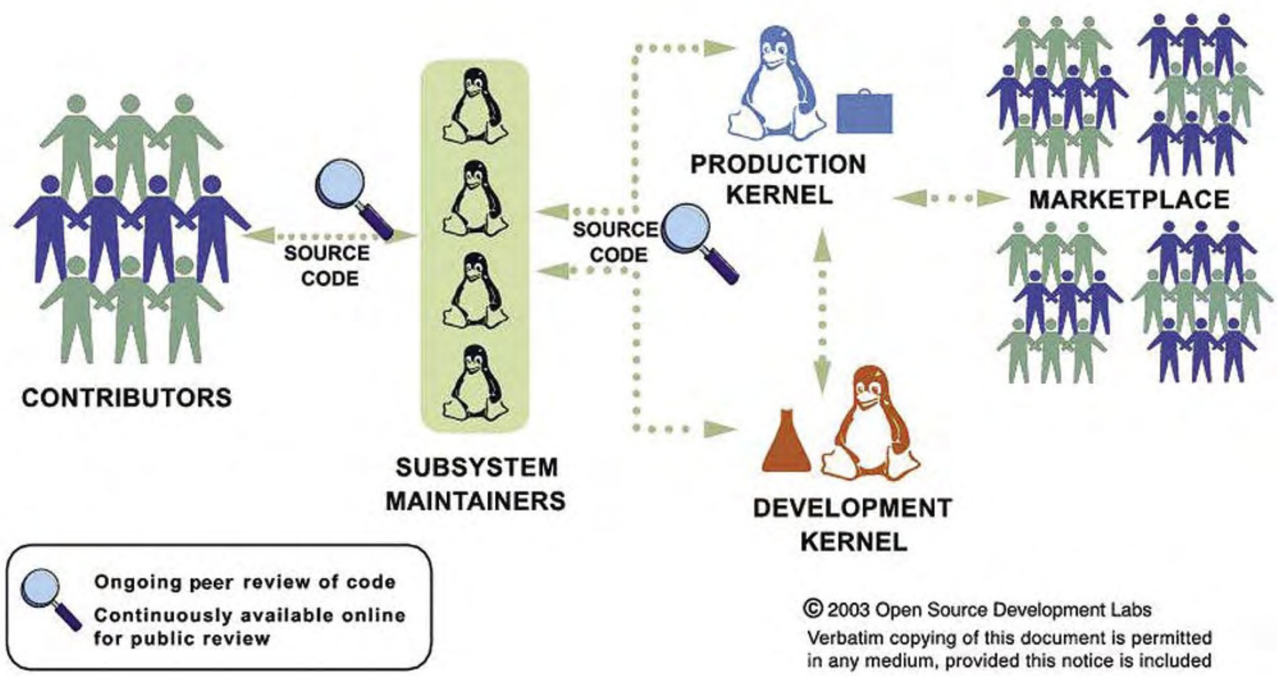


Figure 1

Advancing versions of these subsystems are rolled up into patch sets that ultimately form experimental kernel versions (in the past, odd numbered kernel releases like 2.3.x and 2.5.x). When Linus Torvalds and his team at kernl.org are convinced that Linux kernel technology has advanced and matured sufficiently for commercial deployment, a new production kernel is born and handed off for testing to production kernel maintainers (for example, Marcelo Tosati for 2.4 and Andrew Morton for 2.6). It is normally from these stable production kernels that distribution suppliers build their GNU/Linux OS platforms. For the enterprise: Red Hat, SuSE, and others. For the embedded domain: MontaVista, TimeSys, Wind River, and others.

OSS code origins

Let us examine two origins of Open Source code:

- Open Source code from new projects
- Open Source code integration into mature projects

Open Source code from new projects

Open Source software projects are initiated in the same manner as proprietary projects – from a developer's inspiration, or more formally, in response to requirements expressed by and collected from end-users.

OSS departs radically from traditional commercial software in that its inception need not take place within the confines of a traditional engineering organization. Sometimes new code is written from scratch, and other times, a piece of existing code is re-licensed under an Open Source license and released as a community project.

Some OSS projects, such as applications or middleware, are entirely free-standing. Others exist as patches to the Linux kernel or other OS components – usually device drivers, I/O subsystems, alternate schedulers, and other systems components. A project can exist indefinitely as an independent effort, and is useful as long as its maintainers update it to support current libraries and kernel versions.

If a project shows exceptional merit and broad utility, it can also be picked up by the Linux kernel maintainers and be integrated into the mainstream OS. Such examples include the adoption of the National Security Agency (NSA) Secure SE Linux as a standard build option in the 2.6 Linux kernel, and the incorporation of the Pre-emptible Linux Kernel Patch into the 2.5 and later production kernels. More typical examples are the hundreds of drivers developed by hardware vendors for their devices and later integrated into mainstream kernel trees.

Open Source code integration into mature projects

Larger, mature projects continue to evolve over time and benefit from the contributions of both core team developers as outside submitters. Understanding the path that new submissions follow is instructive for organizations hoping to participate in Open Source

SIDEBAR

OSDL Role in Open Source

The OSDL participates in Open Source development in four distinct capacities.

1

OSDL Initiatives

First, and most visibly, OSDL initiatives, promoted by and for member companies, collect and refine end-user requirements for each of the target areas: Carrier Grade, Data Center, and Desktop Linux. These requirements are mapped against the current capability set of Linux and its stack at any given moment in time with two outcomes: the provision of specifications and capabilities lists to distribution providers, and the fostering of new community development to fill gaps in Linux capabilities.

2

Performance and Regression Testing

Less visibly, the OSDL hosts a massive testing effort for both experimental and production versions of the Linux kernel. This test bed can also be applied to combinations of kernel builds and patch sets using the OSDL Scalable Test Platform (STP) combined with the Linux Patch Line Manager (PLM). OSDL members and associates can upload patches and leverage STP online at the OSDL website.

3

OSDL Member Contributions to Open Source

In response to OSDL initiatives and other organizational imperatives, the over 60 members of the OSDL make substantial contributions to Linux and to other Open Source projects in both enterprise and embedded areas. Founding members include IBM, Intel, HP, CA, Fujitsu, Hitachi, and NEC.

4

Direct Participation in OSS Development

OSDL staff members contribute directly to dozens of Open Source projects such as the Kernel, Asynchronous IO, Persistent Device Naming, Clustering, and Testing. The OSDL also compensates Linus Torvalds, and kernel maintainer Andrew Morton.

development, and for companies concerned about the integrity of the process (Figure 2).

Newcomers to Open Source often speak of *pushing* code out to Open Source, but push is an inappropriate verb to describe the submission process. Individual contributors or organizations actually find it quite challenging to have their patches accepted into actual projects and distributions.

To begin with, a patch must be *well-formed*, which means coded and packaged per the established Open Source conventions, and per the requirements of the given project. Moreover, a patch must be of sufficient merit or novelty to stand out from the dozens (or hundreds) of other proposed patches in front of a project maintainer.

Once a submission actually makes it into a project, it will be exercised, tested, and scrutinized by that project's user community. If it survives and becomes part of a project's mainstream code, and the project is then picked up by a distribution, it will be subject to the integration, testing, and QA discipline that comprises that distribution's added value.

It should be noted that end-users have complete project control, because the inclusion of each specific project package in the distribution into their project is entirely discretionary. Because you have the project source code, you can always return to the project that accepted and built the code if the need arises.

Open Source code quality assurance

Quality assurance for Open Source code occurs at several points in the code lifecycle, and at multiple places in the Open Source ecosystem. As described above, there exist three layers of purview over code submission and quality:

- Project
- Distribution
- End-user

At the project level, there is project-specific testing (including build-testing) that is carried out as part of the project lifecycle. At the same time, the project community joins with the project audience to engage in usability and performance testing.

Project code integration into standard distributions is subject to both standards-based and supplier-specific testing, and the QA that each distribution team performs. In my own experience at an embedded distribution vendor, we performed standards-compliance testing (for example, LSB and POSIX), stability and robustness testing, real-time response and throughput benchmark testing, installation testing, and several other tests and QA checks. We also *cross-pollinated* by comparing test results among our eight supported architecture families.

The result of open source cooperation

If you extend the test and QA process cooperation out to several dozen distribution suppliers, add the bug reports from hundreds of thousands of embedded and enterprise users and developers, and then add the rapid repair and integration that is the hallmark of Open Source, you have a virtualized and borderless QA team that outpaces and outscales even the largest proprietary software organization.

Bill Weinberg *brings more than 17 years embedded and open systems experience to his role as Open Source Architecture Specialist at the Open Source Development Labs. Bill can be contacted at bweinberg@osdl.org.*

OSDL *– home to Linus Torvalds, the creator of Linux – is dedicated to accelerating the growth and adoption of Linux in the enterprise. Contact the OSDL directly for membership and lab usage information.*

Open Source Development Labs, Inc.
12725 SW Millikan Way • Suite 400 • Beaverton, OR 97005
Tel: 503-626-2455 • Fax: 503-626-2436
Website: www.osdl.org

CODE SUBMISSION AND AUDIT PATH

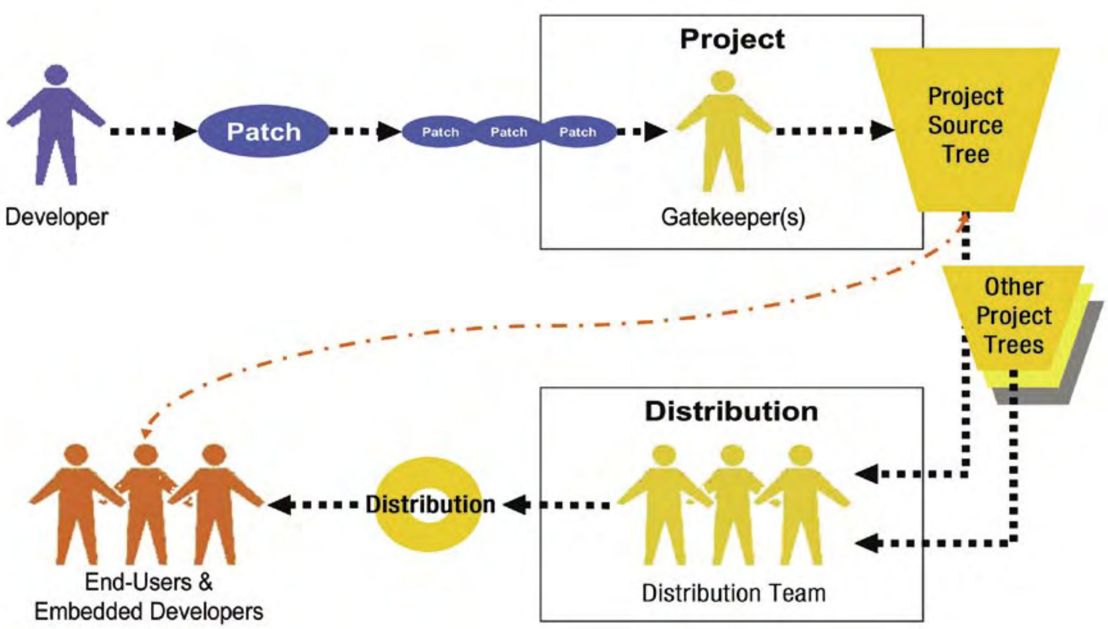


Figure 2

By Ian Mackintosh

Open Core Protocol International Partnership (OCP-IP):

Leading the way in industry openness and collaboration

OCP-IP overview

The Open Core Protocol International Partnership (OCP-IP) is a nonprofit organization delivering the first fully supported, openly licensed core-centric protocol that comprehensively fulfills system-level integration requirements. OCP-IP was announced in December 2001 to promote and support the Open Core Protocol (OCP) as the complete socket standard that ensures rapid creation and integration of interoperable virtual components. The OCP facilitates IP core reusability, reduces design time and risk, and reduces manufacturing costs for SoC designs.

OCP allows designers to build cores independent of specific bus protocols, and of any particular design implementation. This allows easier reuse of OCP-compliant cores across multiple SoC designs. OCP eliminates the need to repeatedly modify the core itself, and preserves the verification and test benches by defining all of the core's natural interface capabilities. They are therefore presented in an unchanging, universally understood manner.

The OCP-IP Governing Steering Committee participants are:

- Nokia
- Sonics Inc.
- STMicroelectronics
- Texas Instruments
- Toshiba Semiconductor Group

The group rapidly exceeded 100 members including IP companies, integrated device manufacturers, system companies, and design houses. OCP-IP has also initiated strategic alliances with several other industry standards organizations including:

- ECSI
- Si2
- OSCI
- VSIA

VSIA endorses the OCP socket, and OCP-IP is an Adoption Group of the VSI Alliance. The success of OCP is a result of the OCP definition of sockets.

Socket definition

For decades, Local Area Networks (LANs) grappled with the same issues that are now emerging for SoC designers. In the end, LAN designers created well-defined interfaces by defining physical connections and protocols for exchanging information over those physical connections. The appearance of these industry conventions enabled the computing industry to provide independently developed and functionally diverse plug-and-play products that commercial enterprises assembled into highly custom LAN configurations. So, the successful implementation of a widely accepted interface definition is not without precedent.

OCP uses the interface concept of *SoC sockets*. Ideally, a SoC socket enables core designers to concentrate on their core functionality and the associated interconnects (for example, USB, 802.11b, or SDRAM). Similarly, SoC system integrators should be able to concentrate on SoC timing, core service bandwidth, latency requirements, and final floor-plan design independent of core functionality. The socket would therefore provide the necessary physical and exchange protocol delineation necessary to achieve this well-defined layering.

Core transport independence

To achieve this, note that an ideal SoC socket must be transport implementation agnostic (in effect, not dependent upon a specific bus or other interconnect). SoC cores therefore interface to an inter-core transport mechanism via the interface, but the precise transport mechanics (such as computer-style bus, a cross bar, or a configurable on-chip network) would be unknown to the core. For example, an IP core with an OCP conformant interface is bus independent as shown in Figure 1.

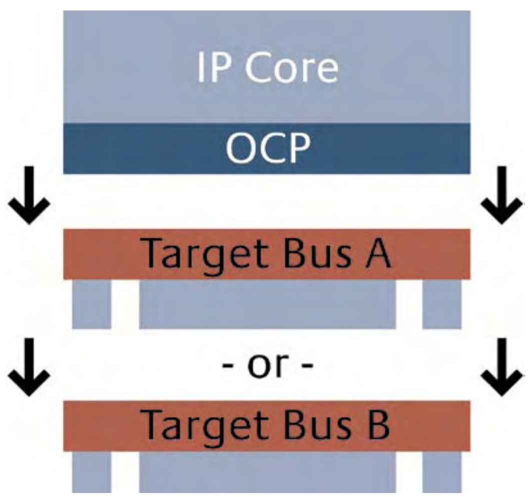


Figure 1

This requirement is essential, or core designs would instantiate transport knowledge within their designs, thereby limiting their reuse in SoC designs that use differing transport mechanics. A transport-unaware approach ensures implementation independence, which allows a system designer to select the optimum interconnect for their system's needs.

Dimension independence

Because of bandwidth requirement diversity, the ideal interface should allow designers to configure interface implementations along various dimensions. For example, these dimensions include the interface data widths required to meet bandwidth requirements, exchange handshake protocols, and exchange acknowledgements. This enables SoC designers to tailor core and SoC designs with minimized complexity and circuit areas, while supporting core and SoC requirements.

OCP benefits

The solution to maximizing core reuse potential requires adopting a well-conceived and specified core-centric protocol as the native core interface. By selecting an adopted industry standard, core designers not only ensure core reuse for cores developed within their own enterprise, they also enable reuse outside their enterprise under Intellectual Property (IP) licensing agreements. Finally, they also maximize their ability to license and incorporate third-party IP within their own SoC designs. In other words, they achieve SoC design agility, and the ability to rapidly implement design solutions when licensing IP.

In addition, a rigorous IP core interface specification, combined with an optimized system interconnect, allows core developers to focus on developing core functions. This eliminates the typical advance knowledge requirements regarding potential end-systems that might utilize a core, as well as the other IP cores that might be present in the application(s). Cores simply need a useful interface that decouples them from system requirements. The interface then assumes the attributes of a socket – an attachment interface that is powerful, frugal, and well understood across the industry.

By this methodology, system integrators realize the benefits of partitioning components through layered hardware – designers no longer have to contend with a myriad of diverse core protocols and inter-core delivery strategies. Using a standard IP core interface eliminates the need to adapt each core during each SoC integration, allowing system integrators the otherwise unrealized luxury of focusing on SoC design issues. And, since the cores are truly decoupled from the on-chip interconnect, hence each other, it becomes trivial to exchange one core for another to meet evolving system and market requirements.

In summary, for true core reuse, cores must remain completely untouched as designers integrate them into any SoC. This can only occur when a change in bus width, bus frequency, or bus electrical loading does not require core modification. In other words, a complete socket insulates cores from the vagaries of, and change to, the SoC interconnect mechanism.

The existence of such a socket enables supporting tool and collateral development for protocol, checkers, models, test benches, and test generators. This allows independent core development that delivers plug-and-play modularity without core interconnect rework. Additionally, this allows core development in parallel with system design thereby saving precious design time.

Core interface requirements

Core interface design requirements are very diverse, and no single specific implementation can possibly address all of the requirements. Standardized core interface specification requires:

- Error handling
- Interrupts
- Test
- More than data-flow signaling
- Control and status
- Flags and software flow control
- Scalability across a family of requirements
- Capture all signaling between the core and the system
- Ability for designers to configure specific interface instantiations along a number of dimensions (such as bus width and data handshaking)

OCP introduction

OCP is a freely available, bus-independent protocol that supports all core-centric considerations discussed previously. Specifically, it completely captures all of an IP core's communication requirements. As a highly configurable interface, OCP is not a one-size-fits-all protocol. Rather, it comprises a continuum of protocols that share a common definition.

OCP explicitly supports sideband signals via optional extensions to the basic OCP data set. These sideband signals include: reset,

interrupt, error, and control/status information. In addition, a generic flag bus accommodates any unique core signaling needs. An optional OCP test interface extension supports scan, JTAG, and clock control. This enables core debug and manufacturing test when integrated into SoCs.

System designers can therefore tailor a specific OCP configuration to exactly match their core requirements. Through straightforward configuration procedures, OCP supports simple, low-performance cores with very simple and frugal OCP interfaces, while also supporting complex, high-performance cores with more complex interfaces.

An IP developer can therefore complete an IP core design using the OCP interface. No end-application knowledge is required beyond the OCP, allowing complete independence for members of the often global design teams. The system integrator is also free to choose the on-chip interconnect that best suits the system requirements of the application, then effectively *wraps* that interconnect to present OCP interfaces to the cores.

Conclusions

A standard socket core protocol is essential to the SoC design community. OCP is the only complete, fully supported, and proven socket. Adopting OCP avoids incompatible or proprietary solution proliferation, and expands the total available market for commercial and legacy IP cores.

The complete, fully supported core-centric OCP delivers substantial and demonstrable benefits over older style bus-centric protocols. OCP is a core-centric, openly licensed, royalty-free core interface protocol. It does not restrict or otherwise interfere with inherent core capabilities. It is scalable and configurable to match different communication requirements associated with different core and SoC designs.

Cores with OCP interfaces and OCP interconnect systems enable true modular plug-and-play integration, allowing the system integrators to optimally choose cores, and the best application interconnect system. This ensures the designer that the cores and the system can work in parallel, and therefore shorten design times. In addition, not having system logic in the cores allows the cores to be immediately reused with no additional time for core modification and reverification.

Finally, verification and test suites, when written to OCP specifications, are completely portable across multiple designs, rarely requiring even minor adjustments for a particular interface bridge.

OCP-IP membership benefits

Website benefits:

- Access to the Members Only website
- Access to Members Discussion Forum
- Free vendor listing
- Free IP and EDA product and design service listing
- E-mail Technical Support

Other benefits:

- Hot line technical support
- Industrial grade tools access
- Free access to all membership products, tools, and services
- Participation in OCP-IP events, work groups, and member meetings
- Ability to associate with key industry leaders in the EDA/SoC community
- The ability to contribute to OCP enhancements and access specifications in both draft and adopted form
- Free training material and tutorials for both the OCP Standard and industrial grade checkers and software tools

Membership applications, and the OCP Specifications are available at the website shown below.

For more information, contact Ian at:

OCP International Partnership
5440 SW Westgate Drive, Suite 217
Portland, Oregon 97221 USA
Tel: 503-291-2560 • Fax: 503-297-1090
E-mail: admin@ocpip.org • Website: www.ocpip.org

Advancing Transaction Level Modeling: Linking the OSCI and OCP-IP Worlds at Transaction Level

By James A. Colgan, Sonics Inc. and Pete Hardee, CoWare, Inc.

Abstract: The growing need for Transaction Level Modeling (TLM) standards that can link together SoC architecture and software development at levels of abstractions higher than RTL has stimulated CoWare and Sonics to collaborate on a cohesive methodology that addresses both SoC designer and software developer needs. This paper outlines the two prevalent industry approaches from OCP-IP (Open Core Protocol – International Partnership) and OSCI (Open SystemC Initiative), and then describes how two founders of both of these organizations are collaborating to provide real solutions to the industry. CoWare is an OCP-IP Sponsor member and founder of OSCI. Sonics is a co-founder of OCP-IP and contributor to the OCP-IP System Level Design Working Group.

Available at: www.opensystems-publishing.com/whitepapers

SHAKE, RATTLE AND ROLL ON.

USP IIIi™ VMEbus COMPUTER

Industry's highest performance VME engine
Single/Dual UltraSPARC® IIIi 64-bit processors
2-way SMP, 64-bit Solaris™ 8 and 9 support
Ruggedized for up to 40G shock

USPIIe-Gb™ VME COMPUTER

650 MHz 64-bit UltraSPARC IIi+ processors
Single/2-slot VME configurations
4GB ECC SDRAM memory, 4 PMC slots
64-bit Solaris 8 and 9 support
Ruggedized for up to 30G shock

TGA3D+™ GRAPHICS ADAPTER

USPIIIi/USPIIe-Gb 3-D graphics accelerator
Provides Sun® XVR-500 3-D graphics support
8.5M triangles/sec, 160M textured pixels/sec
Solaris 8 and 9 support
Sun OpenGL for Solaris 1.2.1 and later

RUGGED, MISSION-CRITICAL COMPUTERS

PROCESSOR PLATFORMS

SUN®
IBM®
INTEL®

OPERATING SYSTEMS

SOLARIS®
LINUX®
WINDOWS®

When it comes to extreme operating conditions, nobody keeps mission-critical applications rolling like Themis Computer.

For over a decade, Themis has delivered high performance, high availability computing for the harshest, most demanding applications and operating environments.

Designed to perform in conditions where other systems fail, our unique thermal and mechanical technology continues to set the standard for rugged, mission-critical computing – up to 40G shock.

Themis' family of UltraSPARC VME and Compact PCI single board computers and graphic accelerators are the latest demonstration of our design expertise, offering ruggedized computers with the fastest, widest range of performance options. Unlike others, our rugged VMEbus computers provide more scalability, far greater reliability, improved life cycle management and substantially lower TCO.

So when your data is critical and the environment is hostile, turn to the computers built to perform when there's a whole lot of shaking going on.

Themis rugged, mission-critical computers. Designed to take it.

THEMIS COMPUTER
3185 Laurelview Court
Fremont, CA 94538
510-252-0870

THEMIS COMPUTER EUROPE
5 rue Irene Joliot-Curie
38320 Eybens, France
+33.476.14.77.86

THEMIS COMPUTER

WWW.THEMIS.COM

Eclipse: The development system that crosses RTOS boundaries

By Robert Day

In the embedded software world, the choice of real-time operating system has typically dictated the choice of development tools. In the 80's and 90's, close relationships were established between RTOS vendors and tool providers (for example, Microtec Research and Software Components Group with XRAY and pSOS).

At the end of the 90's, industry consolidation resulted in major RTOS vendors and device providers owning their own tools technology. With the increase in software in every new embedded design, the reuse of software is becoming critical, making it very unattractive to keep switching embedded software tools.

This leaves us in an interesting tools conundrum. Which tools do you choose if you are using a given RTOS and device combination? And how can you protect your investment in tools (and code generated using those tools) if you switch your RTOS and/or devices? This is further exacerbated if more than one RTOS or devices are used in the same design – a common occurrence in the System-on-Chip (SoC) era.

Are standards the solution?

Standards are often a good way of helping with code reusability. The widespread use of ANSI C as a development language across embedded systems gives developers a chance for reuse. However, there are no real standards for tools and real-time operating systems. Therefore, each design has to use a specific compiler, which generates code for a specific device, to work with a specific RTOS. The RTOS and the tools have a proprietary API from compiler flags, through debug GUI, through IDE interface. This means that there is a large amount of relearning and retooling per project.

What is really needed is an environment that is standard across all embedded systems tools. This would allow the tools and RTOS vendors to be able to plug and play with each other, and give users the benefit of a single tool to manage their projects and code. The environment would also provide standard interfaces and interoperability across the multiple tools and operating systems that are available today. This same environment could also support embedded projects that use a proprietary RTOS, or no RTOS at all.

One of the issues of creating such an environment touches on embedded systems

politics. If either a device company or an RTOS company created this environment, it would become very difficult for their competitors to embrace this as their standard, as it would leave them somewhat at the mercy of their competitor. So having an environment developed and maintained by an agnostic third party is very appealing.

Looking to the desktop or enterprise world is an interesting exercise, as it inherits the benefits of a huge developer network, and the ability to use desktop productivity tools (such as standard editors and version management systems). The problem with this approach is that the desktop world is very host specific (for example, Microsoft Visual Studio). In addition, if embedded features were required, the desktop tools providers would not be readily open to develop those features as they would have a relatively small base of embedded customers.

The Eclipse solution

The good news for the embedded world is that a solution exists. While it may not be well known in the embedded world yet, the solution known as Eclipse is set to revolutionize the embedded software developer's environment. It is as much a culture as it is a product, and it has been embraced by major embedded solutions providers as their standard tools environment.

Eclipse is an open platform for tool integration built by an open community of tool providers. To quote the Eclipse website: "The Eclipse Platform is an open IDE for anything, and for nothing in particular." It was developed by IBM and first released in 2001. In 2004, it was spun out of IBM into a nonprofit corporation called the Eclipse Foundation.

> "While it may not be well known in the embedded world yet, the solution known as Eclipse is set to revolutionize the embedded software developer's environment. It is as much a culture as it is a product..."

The technology is an open framework that is available in source code. The framework is written in Java, and is highly portable across host environments. To date, it is available under Windows, Linux, Solaris, HP-UX, Mac-OS, and IBM AIX.

Eclipse plug-ins

A key factor that makes Eclipse useful is the notion of plug-ins. Each tools provider can build their tools to a certain set of rules and APIs that allow them to *plug in* to the Eclipse framework. In the embedded world, this enables embedded tools providers to build true embedded products that will simply plug into the IDE, and allow them to focus on their core competencies without worrying about developing IDEs.

If the Eclipse plug-in rules are adhered to, the embedded tools will track the latest versions of the Eclipse framework, as well as plug into other Eclipse-based environments.

Eclipse licensing

Another key factor that makes Eclipse useful is the licensing model under which the Eclipse framework is provided. The Common Public License (CPL) provides royalty-free source code and world distribution rights, and allows tools developers to offer the Eclipse framework and their plug-in products without putting their own Intellectual Property (IP) back into the community.

This makes for a very viable business by allowing an open source framework to be a common vehicle, with a common look and feel. The open source framework can also be maintained by an RTOS- and device-agnostic organization, and contain detailed embedded functionality provided by the embedded companies.

Eclipse user advantages

Many of the embedded RTOS companies are now providing an Eclipse-based solution. This means that the embedded user will have a common platform for the common development functions such as project management, editing, file navigation, and build management. The user will also have a common interface to compilation and debugging tools.

If the RTOS providers have implemented their plug-ins correctly, then this one environment can host multiple operating systems without having to change environments. Each RTOS vendor can

provide tools that will have a common Eclipse look and feel, but with specific features that help build and debug applications using that RTOS.

An example of where this is particularly beneficial is in companies with a large product portfolio that serve the same market, but necessitate different user requirements – such as in cell phones.

Cell phone example

High-end cell phones may need a PDA-like operating system such as Palm, Windows CE, or Symbian – whereas middle and low-end phones may use more of a true embedded RTOS like Nucleus PLUS. Much of the software IP will be the same, so having it under a single project manager is very beneficial. The devices are also likely to be the same, so the compilation system can be consistent. Only the RTOS environment and high-end applications will change, and Eclipse can facilitate this.

The cell phone example is also key when considering a multiple-core architecture, as most cell phones use at least one standard processor core, and at least one DSP. Eclipse can host the different compilers, debuggers, and RTOS choices for each processor in one environment.

Perspectives

Eclipse operates using the notion of perspectives, as do engineers. When building, engineers use a build perspective, and when debugging, they use a debug perspective. A nice feature of Eclipse is that plug-ins can cross perspectives. For example, when debugging, the editor and project manager can be made visible and used to make changes to the code if bugs are found, and to navigate around the project to fully understand the context of the debugged code. An example of an Eclipse debug perspective is shown in Figure 1.

Non-embedded plug-ins

Another advantage with Eclipse is the ability to plug in development productivity tools that are not specific to embedded systems. Currently available Eclipse plug-ins provide the following functionality:

- Modeling
- Bug tracking
- Code generation
- Graphics/Drawing
- Source analysis/Testing
- Project/Team management
- Source control (such as CVS, ClearCase, and SourceSafe)

Eclipse provides many advantages for embedded developers; it is now in the hands of the embedded solutions providers to make it a reality.

Nucleus EDGE

At Accelerated Technology, we decided to change our existing development tools environment in 2002. We had based our last generation of tools on Microsoft Visual Studio, as it had offered the only available standard at the time. However, it was proving to be difficult to meet the emerging group of embedded developers who wanted true cross platform support (including Linux host support) and a multi-core embedded development environment.

We took note that Eclipse was emerging as a standard in the enterprise world that offered us a good technical solution and observed that the Eclipse business model allowed us to offer a complete solution to our embedded developers. Nucleus Embedded Developers Graphical Environment (EDGE) utilizes the latest Eclipse platform (version 3), and realizes all of our embedded technology as Eclipse plug-ins.

We have now developed a number of true embedded tools that interoperate under the Eclipse framework. They provide a

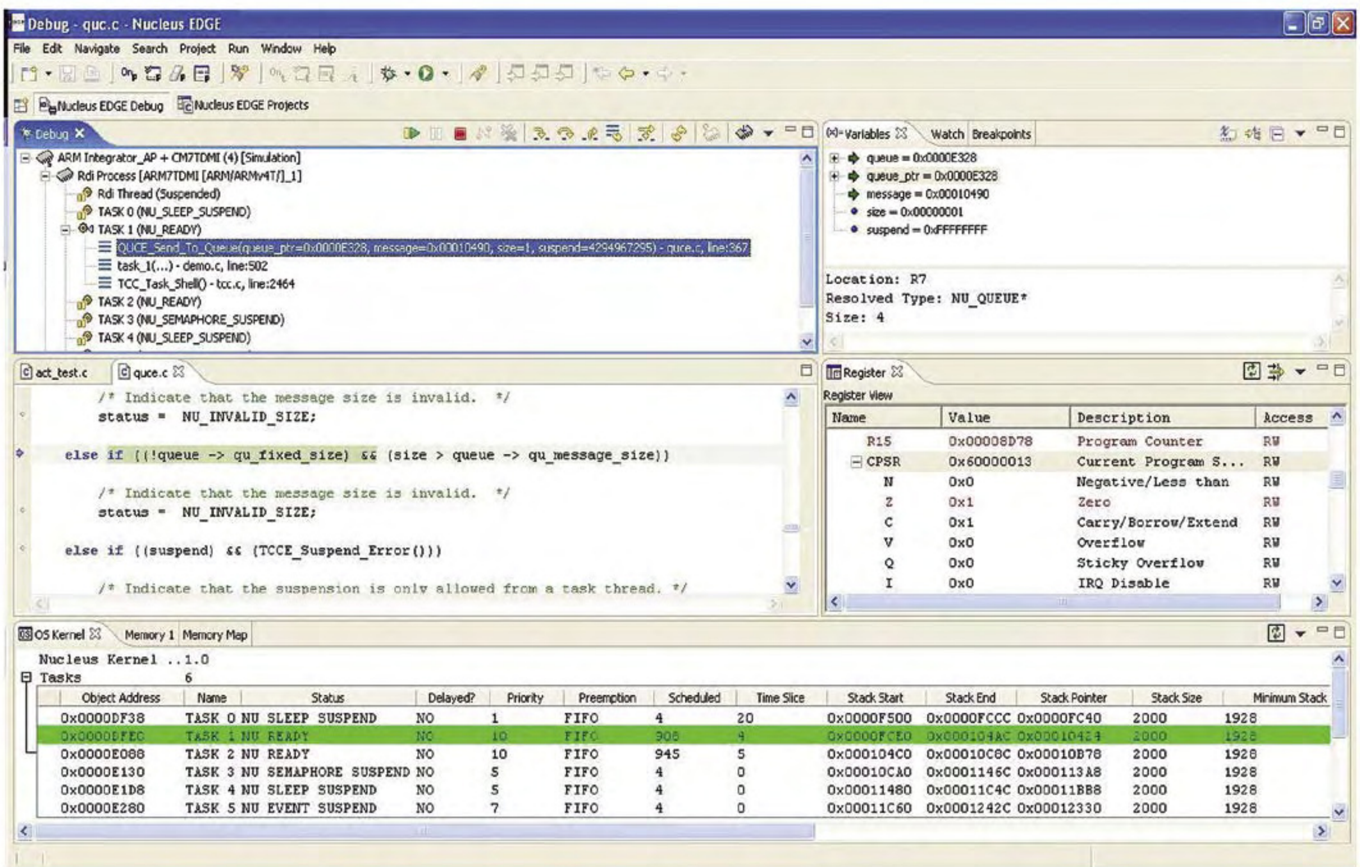


Figure 1

complete development suite for embedded developers who are using Nucleus software, their own proprietary RTOS, or no RTOS at all (see Figure 2).

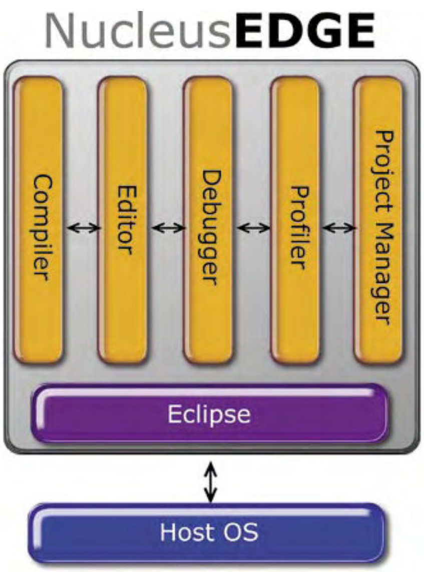


Figure 2

This technology is based on two decades of embedded debuggers, compilers, and RTOS tools. We have added a new project manager interface that complements the Eclipse Navigator view. This allows embedded developers with multiple projects or cores to easily navigate and build their code.

We have also provided our own context sensitive editor that interacts with the build and debug interface, providing a powerful and consistent coding environment throughout the whole software development process. Following the notion of perspectives discussed earlier, this advanced editor is available in both build and debug perspectives, allowing the engineers to make swift code changes as they find their bugs.

Cross compilers are a very target specific link of the embedded software tool chain. Although code is generally written in ANSI C, the compiler is the last part of the tool chain that engineers are ready to change. Within Nucleus EDGE, we offer a number of compiler options that allow users to select their favorite compiler and still receive all the benefits of Eclipse. We offer GNU cross compilers, processor vendor compilers (for example, the ARM RealView compiler), and our own compiler (Microtec compilers). This also allows users of new device architecture to select which compiler is best for their application.

Embedded debugging is a vital part of the software lifecycle. In Nucleus EDGE, we provide many options while using the debug perspective to best accommodate the embedded engineer's debug environment. We offer a native debugging environment (and native compiler to complement it), which allows users to build their applications to run on the architecture of their host machine (for example, PC). This offers rapid software prototyping and is very useful early in the project when hardware is not defined or not available.

The same debug interface also plugs into an Instruction Set Simulator (ISS). This allows the user to compile with the cross compiler but to run on simulated hardware. This is

"Although code is generally written in ANSI C, the compiler is the last part of the tool chain that engineers are ready to change."

digital video
MPEG-2
graphics
analog video
audio
VigraWATCH
The evolution of resolution.
High quality, low bandwidth digital video
VigraWATCH™ is a multi-function, single board COTS solution for real-time video capture, processing and display, and audio input/output. VigraWATCH provides an easy migration path to MPEG-2 video through a reconfigurable FPGA-based architecture combined with an on-board PowerPC, providing:
· MPEG-2 compression/decompression
· Live digital and/or analog video
· High performance and low latency
· Real-time, high resolution graphics display
· Support for off-the shelf and custom algorithms
· Stereo audio input/output, and MPEG-2 audio compression/decompression
· PCI and PMC form factors
We are the leader in the creation of customizable, real-time video and image display solutions. In addition to VigraWATCH, our VigraVision® products have a proven track record of providing high-resolution graphics, overlays and real-time image processing for a wide range of embedded video and imaging applications. Our flexible platforms can handle it all, on-board, without intervention by the host computer.
What's your next system design?
Let our experience go to work for you. Call us.
TITAN
Titan Corporation
Advanced Products & Design Division
boardproducts@titan.com
858.527.6100
www.titan.com
Keyword Search: VigraWATCH
RSC #26 @ www.embedded-computing.com/rsc

very useful to run real system tests if the hardware is not available. The ISS allows detailed examination of code execution and memory, and the debug environment executes this at the source level. The simulation of the clock tick allows the RTOS to also be run at this point, and true system debugging can begin. When target hardware is available, the same debug environment allows connection to the hardware typically through an on-device debug port (JTAG for example).

This debug interface provides a real RTOS-aware view of the system. It provides monitoring of the different RTOS states (such as semaphores, queues, and mailboxes), and information about the state of a particular task or group of tasks. This can be achieved when the system is running (run-mode), or when the system has stopped at a breakpoint (stop-mode). The inclusion of the profiler plug-in can go to the next level, providing full system and timing information down to the task and memory level. This profiler is available as a new perspective, allowing for quick transition from code to system level. The profiler is shown in Figure 3.

An environment that crosses RTOS boundaries

Eclipse and Nucleus EDGE provide a real embedded environment that crosses over RTOS boundaries. Eclipse can offer a solid platform that RTOS vendors can use to build tools to support their RTOS that will have an Eclipse look and feel, and then the embedded developer can take the platform and use whatever RTOS makes most sense for his application without having to change his environment.

Nucleus EDGE provides a proven set of tools that can be used with the Nucleus RTOS, and can also be easily modified to work with another operating system – even a proprietary one. The embedded user then gets the benefit of needing only one set of tools regardless of RTOS. The embedded world is changing for the good of the embedded developer, and Eclipse is helping to fuel the change. **ECD**

Robert Day *is the Director of Marketing for Accelerated Technology, a Mentor Graphics division. Since 1987, Day has held a variety of engineering, sales, and management positions at Mentor Graphics and Microtec Research. Day has more than 15 years embedded industry experience. He holds a Bachelor of Science degree in Computing from the University of Brighton, England.*

For more information, contact Robert at:

Accelerated Technology
739 N. University Blvd.
Mobile, AL 36608
Tel: 251-208-3400 • Fax: 251-343-7074
E-mail: info@acceleratedtechnology.com
Website: www.acceleratedtechnology.com

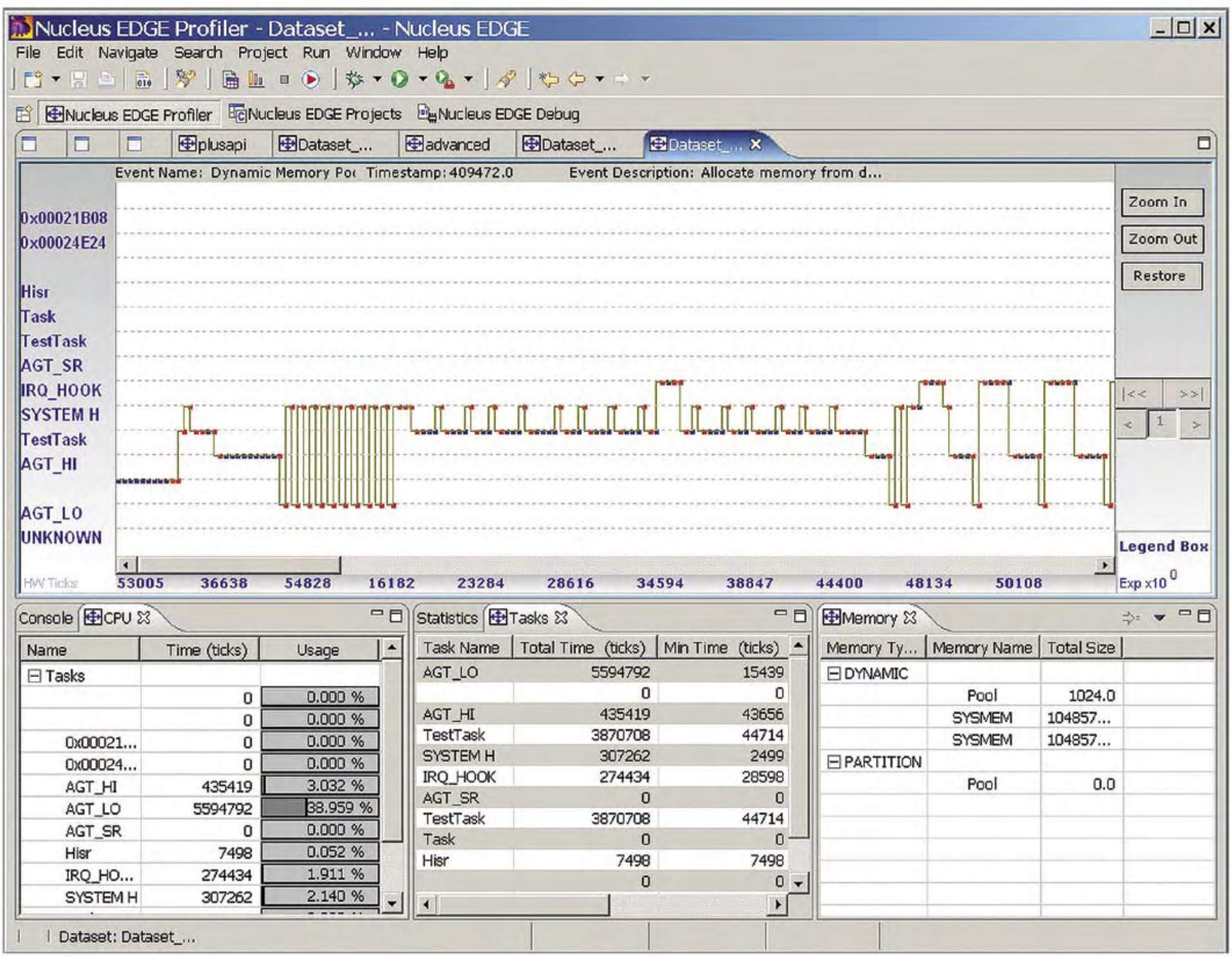


Figure 3

Adding value with the Eclipse framework

By Michael McCullough

The Eclipse framework creates new opportunities and poses difficult design challenges for vendors of Integrated Development Environments (IDEs) in the embedded software industry. The open source nature and extensive built-in capabilities of Eclipse, for instance, force vendors to evaluate exactly how they will provide value, and how they will differentiate their products.

Embedded application developers seeking to maximize their purchasing power and minimize learning curves also face difficult decisions when selecting Eclipse-based products. To date, three distinct approaches have been taken by vendors when incorporating Eclipse into deliverable products. These approaches can be classified as Isolationist, Proprietary, or Truly Open.

Start the evolution without me

To the Isolationist vendor, Eclipse is a threat of the highest magnitude. Eclipse capabilities such as Project tools, File and Directory Management tools, and Language-Aware Editors were key value-adds for tool vendors in the pre-Eclipse world. Throw in the capabilities of the C/C++ Development Tools (CDT) Eclipse plug-in, and now you have GDB-based graphical debugger capability and project management for C and C++; also key value-adds to tool vendors' products.

The collaborative nature of Eclipse flies in the face of Isolationist desires to tie the user to a proprietary environment where the user is often hard-pressed to customize or even change any aspect of the tools. This is why off-the-shelf software development environments are often supplanted at integration time by homegrown approaches.

As this approach refuses to acknowledge the advantages of Eclipse, the remainder of this article will focus on the Proprietary and Truly Open approaches that do utilize Eclipse, and how this affects the development tools offered by vendors.

To be, or not to be, open

With the Proprietary approach, vendors acknowledge the advantages of Eclipse, but are faced with the difficulty of selecting which parts of Eclipse they need to replace to lock in their users. With this approach, the vendor decides or defines exactly which Eclipse pieces will be open and modifiable by the user, and which are not. This allows them to promote the openness granted by Eclipse, while they retain and control their own proprietary capabilities.

The Proprietary approach does free up more of the Eclipse interface to the user. The user can incorporate their own integration

tools and scripting approaches, but they are still essentially locked out from accessing any of the development capabilities provided by the software development environment for their own needs. Work performed during development is simply not reused at integration time. These tools are really only solving half the problem, and are typically only used for half of the actual product development cycle.

The open choice

The remaining vendor approach, Truly Open, acknowledges the extensibility of Eclipse. The vendor offers open, customizable software development and integration tools that are delivered as Eclipse source code. Tools can be used for software development purposes and then scaled appropriately as the product moves into integration and test. This allows a single Eclipse-based interface to cover the entire product development cycle. In the Truly Open model, the user buys once and customizes forever, thereby maximizing their total tools investment.

The impact of using Eclipse

The primary method for vendors to add their own value or integrate their own existing products to Eclipse is via plug-in development. Eclipse offers a fully realized Plug-in Development Environment (PDE) that is not only used to develop plug-ins, it integrates and tests them as well. Once a plug-in has been fully tested, it can be exported to the appropriate Java archive. This archive can then be shipped as a binary-only plug-in that overlays on top of an existing Eclipse installation. Eclipse even offers a plug-in wizard that can be used to create a skeleton or template plug-in that can then be modified to add specific vendor functionality (Figure 1).

In the Proprietary approach, vendors choose what existing Eclipse functionality to replace with their own proprietary substitutes, and they add their own unique additions as well. Replacing existing Eclipse functionality means that the proprietary vendors must now export not only their own plug-ins, but rebuild and re-export several Eclipse plug-ins as well. This puts Proprietary vendors in the unenviable position of distributing their own hybrid version of Eclipse, effectively branching or deviating from the Eclipse open core. Any changes, modifications, or improvements to Eclipse will cause a larger and larger divide between what current Eclipse versions offer, and what the vendor is offering. This is why Proprietary vendors often use older versions of Eclipse, but this widens the gap still more as it can take a year or more to fully productize such a large body of code.

The Truly Open vendor does not have this problem. As their Eclipse product is delivered as full source code, they need only ship the source code to users. The user can now decide what version of Eclipse works best for their needs. Users are not forced to use whatever old, hybrid version of Eclipse the vendor ships and can simply rebuild the tool source code as necessary. Source code control issues are much less problematic for the Truly Open vendor, as locking down the Eclipse environment is now no longer a requirement. The Truly Open vendor may not even need to ship a version of Eclipse at all with their product, so product distribution is simplified as well.

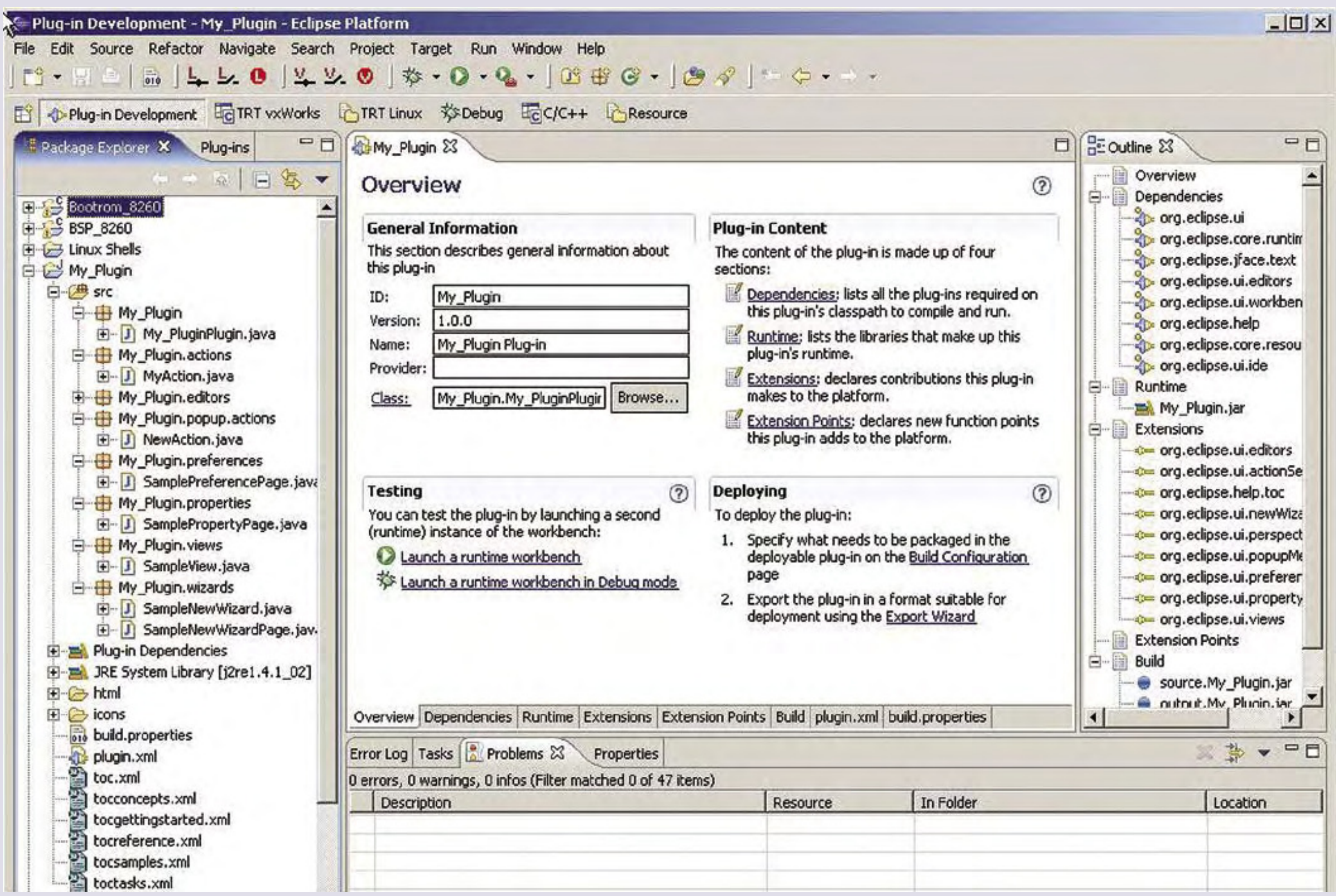


Figure 1

The great editor debate

When developing Eclipse plug-ins, there are several lower level design decisions that can also affect vendors and users. One of the key decisions a vendor faces is in the area of Views and Editors. In Eclipse, each View that is used for interactions with users is not file-like, which means that the user may modify the actual data, but there is no underlying file metaphor. For file-type access or anything that would typically require cut, copy, and paste operations, vendors typically use an Eclipse Editor instead. Unfortunately for vendors, there is a significant difference in complexity between Views and Editors. In addition to providing all the same functionality seen in Views, Editors have considerably more line editor and other requirements as well.

Proprietary vendors have several choices in creating Editors. The vendor can integrate their very own file access with their own editor, can modify the existing Eclipse editors, or can choose not to modify at all and simply reuse the existing Eclipse editors. Proprietary vendors will usually modify an existing Eclipse editor or create their own from scratch. Often, this is because they are creating their own Project Managers, so specific file and directory manipulation is being coded regardless.

> "In Eclipse, each View that is used for interactions with users is not file-like, which means that the user may modify the actual data, but there is no underlying file metaphor."

Truly Open vendors have a large advantage in this debate. Since Truly Open vendors deliver source code for any Eclipse Editor modifications or improvements, the user can decide whether or not to utilize the vendor modifications, reuse the existing Eclipse Editors, or even integrate a third-party editor. Given the amount of time spent debating the usefulness of one editor over another within the user community, it is a distinct advantage not to force a specific editor on a user. Truly Open vendors can even support customer sites where one user uses the vendor's editor, another uses the standard Eclipse Editors, and still another uses a third-party editor. Truly Open vendors simply give the user more choice in this area.

Project management

While the earliest versions of Eclipse were designed specifically for Java development, it was recognized early on that the environment could also be modified to support C and C++ development in an equally open manner. The CDT project was created to do just that, using the GNU compiler tool chain and the gdb debugger. Now there is integrated Eclipse support for Java, C, C++, and other languages as well through third-party vendors. The Eclipse Project Manager was also improved by including management for C and C++ projects through the CDT.

Historically, Proprietary vendors have offered their own proprietary project management capabilities within their own proprietary toolsets. This makes Project Management another area where Proprietary vendors must make a choice similar to that of the Editors. Do they utilize the existing Project Management, modify it, or create their own? To date, most, if not all, Proprietary vendors have chosen to create their own project managers. The only need that is met by this choice, however, is that of the Proprietary vendors for control.

Experienced Eclipse users can modify Eclipse to build just about any command-line project provided that the tools are properly gnu-like. Even non-gnu builds can be accomplished provided they can be executed on a command line with little if any modifications to the standard Eclipse environment. Since proprietary vendors keep their project management closed to the user, the user cannot easily make either site-specific or configuration-specific changes.

Truly Open vendors understand that any modifications to the standard Project Management capabilities are usually unnecessary, and instead choose to document the steps necessary to build for their particular types of projects. If they do offer modifications to the Project Management, the user has the ability to choose which, if any, functionality they will use on a day-to-day basis and only build in the appropriate capability. The Truly Open approach to Eclipse Project Management again simply gives the user more choice.

CDT debugger

The CDT debugger is a straightforward graphical front-end to gdb. It does expect a version of gdb with the mi interface which allows the full use of all of the Eclipse debug capabilities. For embedded developers, this requires a cross-target version of gdb to be built. These gdb versions are available for most, if not all, 32-bit processor architectures or can be developed from existing gdb installations. The CDT debugger is potentially one of the most complex group of plug-ins in Eclipse and can be quite difficult to modify and rebuild.

Many Proprietary vendors are using the complexity of the CDT debugger to their own advantage. As most users simply will not have the time to learn how to modify then rebuild the CDT, these vendors are making their own custom debuggers closed to the user. In many cases these vendors make claims to the unsuitability of the CDT debugger for their particular OS or system, but these are dubious claims at best, especially given the ongoing community improvements to the CDT.

Truly Open vendors have a solution to the complexity issue by delivering fully realized preconfigured projects for rebuilding the CDT. These vendors make their modifications available to the user so that the user can understand the inner workings of the CDT without having to spend too much time reading through a large code base. Truly Open vendors understand that by doing this, they are in fact creating more opportunities for consulting or at least supporting the user down the road. A Truly Open debugger system offers users a degree of tool customization never seen before.

Perspectives

One of the most powerful customization capabilities offered by the Eclipse environment is the ability to group together specific Views, Editors, Project Managers, and Debuggers into a single cohesive unit called a Perspective. Perspectives are managed by Eclipse, allowing multiple Perspectives to exist in the environment at the same time. These Perspectives can then be used for specific development and integration tasks.

Proprietary vendors use this capability to group together their own proprietary functionality along with existing Eclipse capabilities into effectively proprietary Perspectives. This approach often gives

the user the false impression that it is not possible to create new Perspectives using the vendor functionality. This simply is not the case, however. The user can create their own Perspectives with or without the vendor's proprietary functionality as they choose.

Truly Open vendors understand how Eclipse Perspectives are created, and offer the user examples of Perspective creation either built into their deliverable product or documented in user manuals. These vendors let the user decide which Perspectives will or will not suit their specific needs.

Feature projects

Features in Eclipse are a powerful branding tool. They allow vendors to create a custom branded environment where the user often sees only the vendor's infrastructure, thereby effectively hiding the true Eclipse underpinnings. Features allow a grouping of capabilities and Perspectives into a single vendor-specific entity. Feature projects are created in much the same way as standard plug-ins.

Proprietary vendors truly appreciate the Features capability. Features allow Proprietary vendors to encapsulate all the control they choose into a unified whole, presenting a single vendor-customized interface that in many cases will remain immune against modifications or improvements from the user community. While achieving a great solution to the company branding issue, this is not necessarily the best solution for all users.

In many cases, the user may need to publish their own Eclipse environment and desire, much as the Proprietary vendors do, the ability to present a single interface to their end user. While this ability is usually not supported by Proprietary vendors, Truly Open vendors understand this need and have methods to address this. In any case, the user of a Truly Open Eclipse interface can simply remove the branding functionality as the user has total control over what is built into the environment that they will present to their own developers.

The power of Eclipse and open source

The advent of the Eclipse framework and the CDT project presents a series of difficult design decisions to the embedded software development tool vendor. In many cases, functionality that was once proprietary has now been opened up by Eclipse and made available to the user. The approach chosen by the vendor and their underlying design decisions has definitive, measurable consequences for the user, however.

In many cases, the Isolationist and Proprietary vendors, often for essentially marketing reasons, must make design decisions that limit the amount of control that the user will have over the Eclipse environment. This diminishes the open and evolutionary quality of the Eclipse model itself, and usually only to the vendor's benefit. Truly Open vendors have, in many cases, simpler decisions to make concerning the environment itself, thereby giving greater flexibility to the user to decide what is best for the user's development and integration needs.

This simplification is not without consequences for the Truly Open vendor. Vendors adopting the Truly Open approach simply cannot use the same business model as the Proprietary or Isolationist vendors. The most added value that Truly Open vendors offer actually occurs after the tools sale through training and consulting opportunities. Truly Open vendors must have a very active and skilled consultative approach to user support.

Proprietary vendors, because of their closed approaches, simply cannot participate at the level that Eclipse users will expect and prefer. Many users simply will not have the time or the personnel to learn how to make their own improvements or modification to the tools environment and will need expert and cost-effective guidance in this effort. Truly Open vendors capitalize on this user need by giving the user what the user has always wanted: total control over the software development environment, and skilled assistance when and where the user needs it.

Only market demands and time will decide the winner of the Isolationist, Proprietary, or Truly Open contest in the new era of Eclipse-based embedded software development tools. We believe that only the Truly Open approach provides embedded developers with the flexibility and capability they need for the projects they will undertake today, and in the future. Long established tool vendors will need to reevaluate their business models and

"...functionality that was once proprietary has now been opened up by Eclipse and made available to the user."

marketing approaches, or risk losing an established customer base to the powerful draw of the open source software model and those vendors who support it. ECD

__Mike McCullough__ is President and CEO of MCC Systems, Inc. Mike has a BS in Computer Engineering, and an MS in Systems Engineering from Boston University. A 20-year electronics veteran, he has held various positions at Wind River Systems, Lockheed Sanders, Stratus Computer, and Apollo Computer.

For more information, contact Mike at:

MCC Systems, Inc.
9 Cheshire Court
Londonderry, NH 03053
Tel: 603-537-9593 • Fax: 831-309-3472
E-mail: mike@mccengineering.com
Website: www.mccengineering.com

RTOS versus GPOS: What is best for embedded development?

By Paul N. Leroux

Do most embedded projects still need an RTOS? It is a good question, given the speed of today's high-performance processors and the availability of real-time patches for Linux, Windows, and other General Purpose Operating Systems (GPOSs).

Speed without the expense

The answer lies in the very nature of embedded systems, which are often manufactured by the thousands or even millions of units. Even a one dollar reduction in per-unit hardware costs can save the manufacturer a small fortune. Many of these systems are cost sensitive, where the use of multi-gigahertz processors or a large memory array is not possible. In the automotive telematics and infotainment market, for instance, the typical 32-bit processor runs at about 200 MHz – a far cry from the 2 GHz or faster processors now common in desktops and servers. In an environment like this, an RTOS designed to extract extremely fast (and predictable) response times from lower-end hardware offers a serious economic advantage.

Savings aside, the services provided by an RTOS make many computing problems easier to solve, particularly when multiple activities compete for a system's resources. Consider, for instance, a system where users expect immediate response to input. With an RTOS, a developer can guarantee that operations initiated by the user will execute in preference to other system activities, unless a more important activity must execute first (for example, an operation that protects user safety).

Consider also a system that must satisfy Quality of Service (QoS) requirements, such as a device that presents live video. If the device depends on software for any part of its content delivery, it can experience dropped frames at a rate that users perceive as unacceptable. From the user's perspective, the device is unreliable. But with an RTOS, the developer can precisely control the order in which software processes execute, and thereby ensure that playback occurs at an appropriate and consistent media rate.

RTOSs are not fair

The need for *hard* real time – and for OSs that enable it – remains prevalent in the embedded industry. For evidence, consider recent developments in the Linux world. MontaVista, for example, has launched an open source project in an attempt to improve task preemption in the Linux kernel. Meanwhile, a recent study conducted by Venture Development Corporation suggests that lack of real-time performance is the biggest impediment to Linux adoption. The questions are:

- What does an RTOS have that a GPOS does not?
- How useful are the real-time extensions now available for some GPOSs?
- Can such extensions provide a reasonable facsimile of RTOS performance?

Task scheduling

Let's begin with task scheduling. In a GPOS, the scheduler typically uses a *fairness policy* to dispatch threads and processes onto the CPU. Such a policy enables the high overall throughput required by desktop and server applications, but offers no guarantees that high-priority, time-critical threads will execute in preference to lower-priority threads.

For instance, a GPOS may decay the priority assigned to a high-priority thread, or otherwise dynamically adjust the priority in the interest of fairness to other threads in the system. A high-priority thread can, as a consequence, be preempted by threads of lower priority. In addition, most GPOSs have unbounded dispatch latencies: the

more threads in the system, the longer it takes for the GPOS to schedule a thread for execution. Any one of these factors can cause a high-priority thread to miss its deadlines – even on a fast CPU.

In an RTOS, on the other hand, threads execute in order of their priority. If a high-priority thread becomes ready to run, it will, within a small and bounded time interval, take over the CPU from any lower-priority thread that may be executing. Moreover, the high-priority thread can run uninterrupted until it has finished what it needs to do – unless, of course, it is preempted by an even higher-priority thread. This approach, known as priority-based preemptive scheduling, allows high-priority threads to meet their deadlines consistently, no matter how many other threads are competing for CPU time.

Preemptible kernel

For most GPOSs, the OS kernel is not preemptible. Consequently, a high-priority user thread can never preempt a kernel call, but must instead wait for the entire call to complete – even if the call was invoked by the lowest-priority process in the system. Moreover, all priority information is usually lost when a driver or other system service, usually performed in a kernel call, executes on behalf of a client thread. Such behavior causes unpredictable delays and prevents critical activities from completing on time.

In an RTOS, on the other hand, kernel operations are preemptible. There are still windows of time in which preemption may not occur, but in a well-designed RTOS, those intervals are extremely brief, often on the order of hundreds of nanoseconds. Moreover, the RTOS will impose an upper bound on how long preemption is held off and interrupts disabled; this allows developers to ascertain worst-case latencies.

To achieve this goal, the RTOS kernel must be simple and as elegant as possible. Only services with a short execution path should be included in the kernel itself. Any operations that require significant work (for instance, process loading) must be assigned to external processes or threads. Such an approach helps ensure that there is an upper bound on the longest non-preemptible code path through the kernel.

In a few GPOSs, such as Linux 2.6, some degree of preemptibility has been added to the kernel. However, the intervals during which preemption may not occur are still much longer than those in a typical RTOS; the length of any such preemption interval will depend on the longest critical section of any modules incorporated into the kernel (for example, networking and file systems). Moreover, a preemptible kernel does not address other conditions that can impose unbounded latencies, such as the loss of priority information that occurs when a client invokes a driver or other system service.

Avoiding priority inversion

Even in an RTOS, a lower-priority thread can inadvertently prevent a higher-priority thread from accessing the CPU – a condition known as *priority inversion.* Generally speaking, priority inversion occurs when two tasks of differing priority share a resource, and the higher-priority task cannot obtain the resource from the lower-priority task. To prevent this condition from exceeding a fixed and bounded interval of time, an RTOS may provide a choice of mechanisms including priority inheritance and priority ceiling emulation. We could not possibly do justice to both mechanisms here, so let us focus on a simple example of priority inheritance.

To begin, we first must look at the blocking that occurs from synchroniza-tion in systems, and how priority inversion

"Generally speaking, priority inversion occurs when two tasks of differing priority share a resource, and the higher-priority task cannot obtain the resource from the lower-priority task."

can occur as a result. Let us say two jobs are running, and that Job 1 has the higher priority. If Job 1 is ready to execute, but must wait for Job 2 to complete an activity, we have blocking. This blocking may occur as a result of synchronization – waiting for a shared resource controlled by a lock or a semaphore – or as a result of requesting a service.

The blocking allows Job 2 to run until the condition that Job 1 is waiting for occurs (for instance, Job 2 unlocks the resource that both jobs share). At that point, Job 1 gets to execute. The total time that Job 1 must wait may vary, with a minimum, average, and maximum time. This interval is known as the blocking factor. If Job 1 is to meet any of its timeliness constraints, this factor cannot vary according to any parameter, such as the number of threads or an input into the system. In other words, the blocking factor must be bounded.

Now let us introduce a third job that has a higher priority than Job 2 but a lower priority than Job 1 (Figure 1). If Job 3 becomes ready to run while Job 2 is executing, it will preempt Job 2, and Job 2 will not be able to run again until Job 3 blocks or completes. This will, of course, increase the blocking factor of Job 1; that is, it will further delay Job 1 from executing. The total delay introduced by the preemption is a priority inversion.

In fact, multiple jobs can preempt Job 2 in this way, resulting in an effect known as chain blocking. Under these circumstances, Job 2 might be preempted for an indefinite period of time, yielding an unbounded priority inversion, causing Job 1 to fail to meet any of its timeliness constraints. This is where *priority inheritance* comes in. If we return to our scenario and make Job 2 run at the priority of Job 1 during the synchronization period, then Job 3 will not be able to preempt Job 2, and the resulting priority inversion is avoided (Figure 2).

Dueling kernels

GPOSs – Linux, Windows, and various flavors of UNIX – typically lack the mechanisms we have just discussed. Nonetheless, vendors have developed a number of real-time extensions and patches in an attempt to fill the gap. There is, for example, the dual-kernel approach, in which the GPOS runs as a task on top of a dedicated real-time kernel (Figure 3).

Any tasks that require deterministic scheduling run in this kernel, but at a higher priority than the GPOS kernel. These tasks can thus preempt the GPOS whenever they need to execute and will yield the CPU to the GPOS only when their work is done.

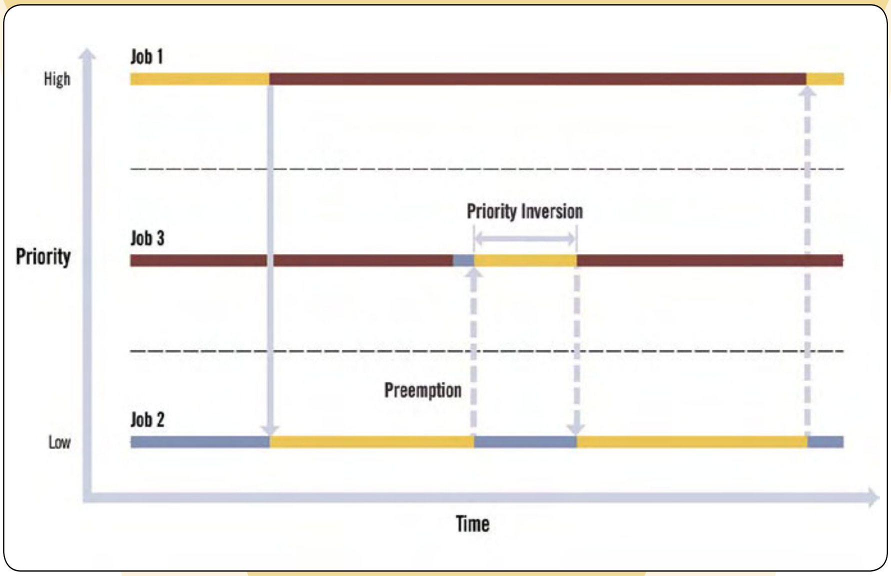


Figure 1

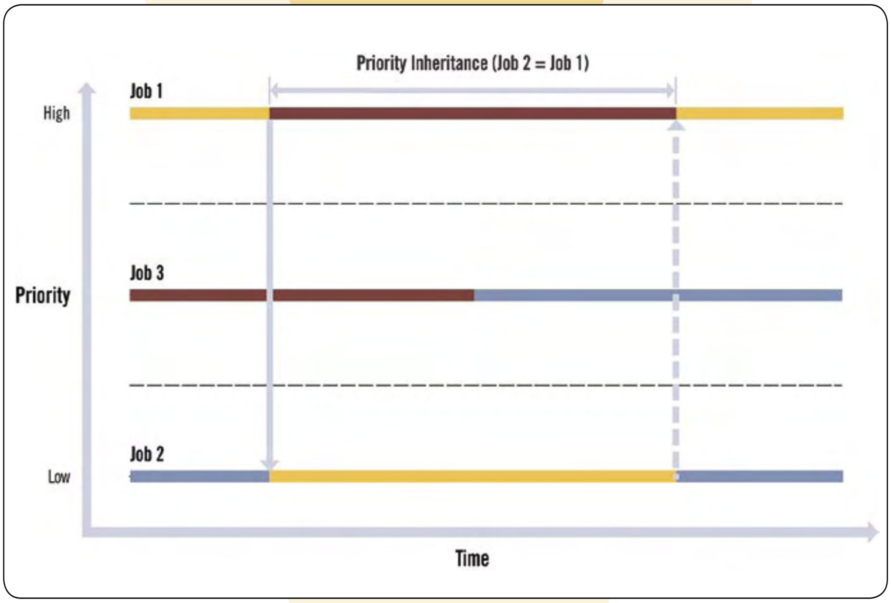


Figure 2

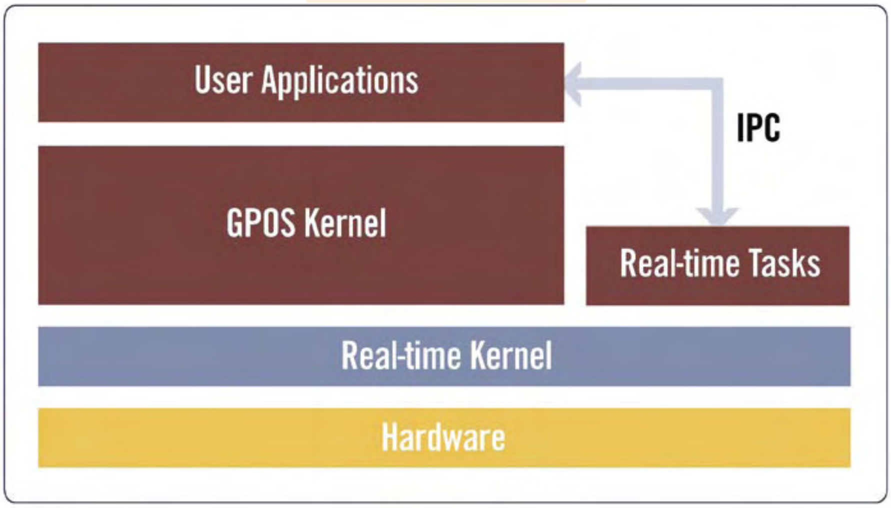


Figure 3

RTOS vs GPOS

Unfortunately, tasks running in the real-time kernel can make only limited use of existing system services in the GPOS – file systems, networking, and so on. In fact, if a real-time task calls out to the GPOS for any service, it will be subject to the same preemption problems that prohibit GPOS processes from behaving deterministically. As a result, new drivers and system services must be created specifically for the real-time kernel – even when equivalent services already exist for the GPOS.

Also, tasks running in the real-time kernel do not benefit from the robust Memory Management Unit (MMU) protected environment that most GPOSs provide for regular, non-realtime processes. Instead, they run unprotected in kernel space. Consequently a real-time task that contains a common coding error, such as a corrupt C pointer, can easily cause a fatal kernel fault. To complicate matters, different implementations of the dual-kernel approach use different APIs. In most cases, services written for the GPOS cannot be easily ported to the real-time kernel, and tasks written for one vendor's real-time extensions may not run on another's real-time extensions.

Modified GPOSs

Rather than use a second kernel, other approaches modify the GPOS itself, such as by adding high-resolution timers or a modified process scheduler. Such approaches have merit, since they allow developers to use a standard kernel (albeit with proprietary patches) and programming model. Moreover, they help address the requirements of reactive, event-driven systems.

Unfortunately, such low-latency patches do not address the complexity of most real-time environments, where real-time tasks span larger time intervals and have more dependencies on system services and other processes than do tasks in a simple event-driven system. For instance, in systems where real-time tasks depend on services such as device drivers or file systems, the problem of priority inversion would have to be addressed.

In Linux, for example, the driver and Virtual File System (VFS) frameworks would effectively have to be rewritten along with any device drivers and file systems employing them. Without such modifications, real-time tasks could experience unpredictable delays when blocked on a service. As a further problem, most existing Linux drivers are not preemptible. To ensure predictability, programmers would also have to insert preemption points into every driver in the system.

All this points to the real difficulty, and immense scope, of modifying a GPOS so it is capable of supporting real-time behavior. However, this is not a matter of *RTOS good, GPOS bad*. GPOSs such as Linux, Windows XP, and UNIX all serve their intended purposes extremely well. They only fall short when they are forced into deterministic environments they were not designed for, such as those found in automotive telematics systems, medical instruments, and continuous media applications.

What about an RTOS?

Still, there are undoubted benefits to using a GPOS, such as support for widely used APIs, and in the case of Linux, the open source model. With open source, a developer can customize OS components for application-specific demands and save considerable time troubleshooting. The RTOS vendor cannot afford to ignore these benefits. Extensive support for POSIX APIs – the same APIs used by Linux and UNIX – is an important first step. So is providing well-documented source and customization kits that address the specific needs and design challenges of embedded developers.

The architecture of the RTOS also comes into play. An RTOS based on a microkernel design, for instance, can make the job of OS customization fundamentally easier to achieve. In a microkernel RTOS, only a small core of fundamental OS services (such as signals, timers, and scheduling) reside in the kernel itself. All other components (such as drivers, file systems, protocol stacks, and applications) run outside the kernel as separate, memory-protected processes (Figure 4).

As a result, developing custom drivers and other application-specific OS extensions does not require specialized kernel debuggers or kernel gurus. In fact, as user-space programs, such extensions become as easy to develop as regular applications, since they can be debugged with standard source-level tools and techniques.

For instance, if a device driver attempts to access memory outside its process container, the OS can identify the process responsible, indicate the location of the fault, and create a process dump file viewable with source-level debugging tools. The dump file can include all the information the debugger needs to identify the source line that caused the problem,

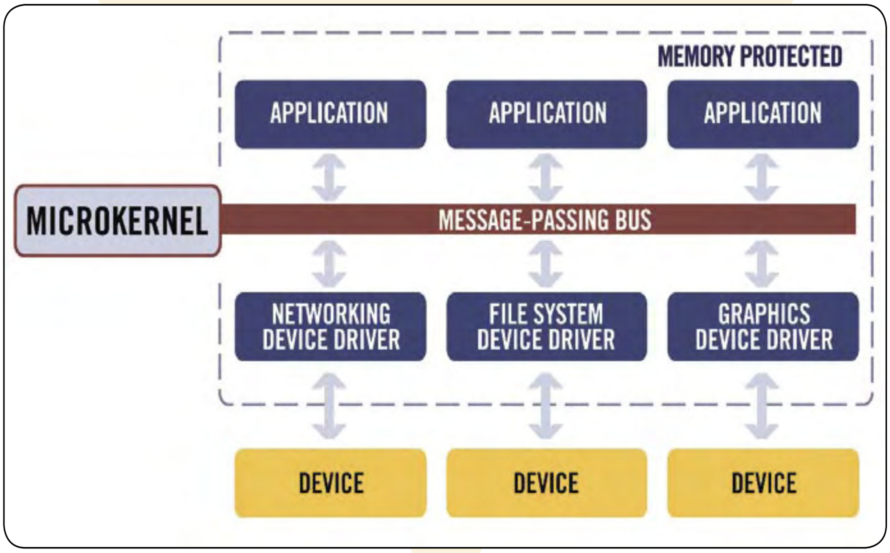


Figure 4


Off-the-shelf XScale® based single board computers and industrial wireless solutions...
Cellular Wireless/GPS
• Embedded baseboard to support low power PC/104 CPU boards
• Wireless cellular modem fitting options:
- Motorola i01500R (iDEN)
- Sony Ericsson GM47/48 (GPRS)
- Siemens MC35 (GPRS)
• GPS module - Trimble LassenSQ
• Wide input PSU: 10 - 30V DC
Get to market faster with Arcom's ready-to-run, modular hardware for GPRS, iDEN and GPS wireless solutions...
VIPER
• 400MHz Intel® PXA255 processor
• Ultra-low power 1.9W (typical)
• 64Mbytes DRAM / 32Mbytes Flash
• Ethernet, USB and 5 x serial ports
• CompactFlash (CF+) and PC/104 bus
All trademarks recognized
MERCURY
• 533MHz Intel® IXP425 comms processor
• 2 x 100baseTx Ethernet ports
• 4 x USB 2.0 and 4 x serial ports
• 64Mbytes DRAM / 32Mbytes Flash
• CompactFlash (CF+) and PC/104 bus
www.arcom.com
Development Kits
Embedded Boards
PCs & Enclosures
Industrial Networking
Design & Build
Arcom
us-sales@arcom.com
888-941-2224
Think Embedded. Think Arcom.

RTOS vs GPOS

along with diagnostic information such as the contents of data items and a history of function calls.

This architecture also provides superior fault isolation. If a driver, protocol stack, or other system service fails, it can do so without corrupting other services or the OS kernel. In fact, *software watchdogs* can continuously monitor for such events and restart the offending service dynamically, without resetting the entire system or involving the user in any way. Likewise, drivers and other services can be stopped, started, or upgraded dynamically, again without a system shutdown.

A strategic decision

An RTOS can help make complex applications both predictable and reliable. In fact, the predictability made possible by an RTOS adds a form of reliability that cannot be achieved with a GPOS (if a system based on a GPOS does not behave correctly due to incorrect timing behavior, then we can justifiably say that the system is unreliable). Still, choosing the right RTOS can itself be a complex task. The underlying architecture of an RTOS is an important criterion, but so are other factors.

Consider Internet support. Does the RTOS support an up-to-date suite of protocol stacks such as IPv4, IPv6, IPsec, SCTP, and IP filtering with NAT? And what about scalability? Does the RTOS support a limited number of processes, or does it allow hundreds or even thousands of processes to run concurrently? And does it provide support for distributed or symmetric multiprocessing?

GUI considerations

Graphical User Interfaces (GUIs) are becoming increasingly common in embedded systems, and those interfaces are becoming increasingly sophisticated. Consequently, does the RTOS support primitive graphics libraries, or does it provide an embeddable windowing system that supports 3D rendering, multi-layer interfaces, and other advanced graphics? Can you customize the GUI's look-and-feel? Can the GUI display and input multiple languages simultaneously? And does the GUI support an embeddable web browser? The browser should have a scalable footprint, and be capable of rendering web pages on very small screens. It should also support current standards such as HTML 4.01, XHTML 1.1, SSL 3.0, and WML 1.3.

Tool considerations

On the tools side, does the RTOS vendor offer diagnostic tools for tasks such as trace analysis, memory analysis, application profiling, and code coverage? And what of the development environment? Is it based on an open platform like Eclipse, which lets you readily plug in third-party tools for modeling, version control, and so on? Or is it based on proprietary technology?

On one point, there is no question. The RTOS can play a key role in determining how reliable your system will be, how well it will perform, and how easily it will support new or enhanced functionality. And it can support many of the rich services traditionally associated with GPOSs, but implemented in a way to address the severe processing and memory restraints of embedded systems. ECD

***Paul Leroux** is a Technology Analyst at QNX Software Systems, where he has served in various roles since 1990. His areas of focus include OS architecture, high availability systems, and integrated development environments.*

For more information, contact Paul at:

QNX Software Systems Ltd.
175 Terence Matthews Crescent
Ottawa, Ontario, Canada K2M 1W8
Tel: 613-591-0931 • Fax: 613-591-3579
E-mail: paull@qnx.com
Website: www.qnx.com

Taking the heat: Strategies for cooling SBCs in commercial and military environments

By Ivan Straznicky and Phuc Nguyen

Cooling high-density electronics is a key issue for today's embedded computing hardware. The bad news is that the problem will only increase in importance with tomorrow's hotter, denser technology. Unfortunately, it is no longer adequate for system designers to simply increase air flow for air-cooled hardware, or put more copper into PWBs for conduction-cooled hardware.

Present and future power/heat levels dictate that sophisticated design, analysis, and testing approaches be used to successfully field products. Strategies for successful cooling of Single Board Computers (SBCs) vary depending on whether the system is to be deployed in an air-cooled commercial environment (industrial or laboratory class), or a conduction-cooled rugged environment (defense and aerospace).

Analyzing the problem

Current SBC and mezzanine modules are either conduction- or air-cooled. Solid technical understanding of heat transfer, along with sophisticated thermal and mechanical modeling and analysis capabilities, is driving continued innovation in both areas to extend their capabilities. An example of a 3-D conduction cooling analysis is shown in Figure 1. An example of a Computational Fluid Dynamics (CFD) analysis on an air-cooled product is shown in Figure 2.

Analyses such as these can be accomplished relatively quickly and easily with today's powerful computing hardware and user-friendly software. The challenge is accuracy of results. This requires an understanding of the many input variables, how they impact the results, and what values to use.

For example, simple use of data sheet thermal conductivity (k) values for interface

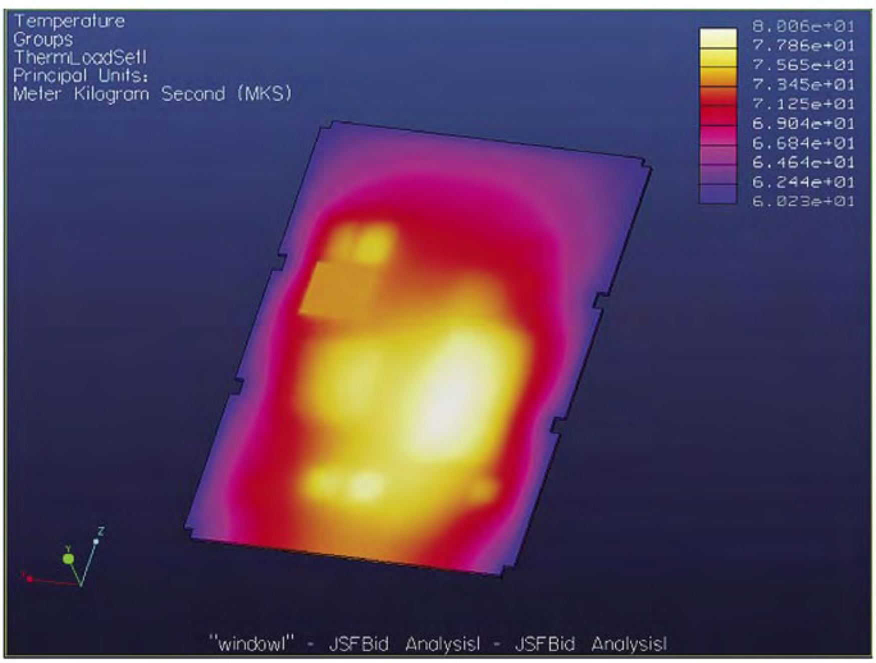


Figure 1

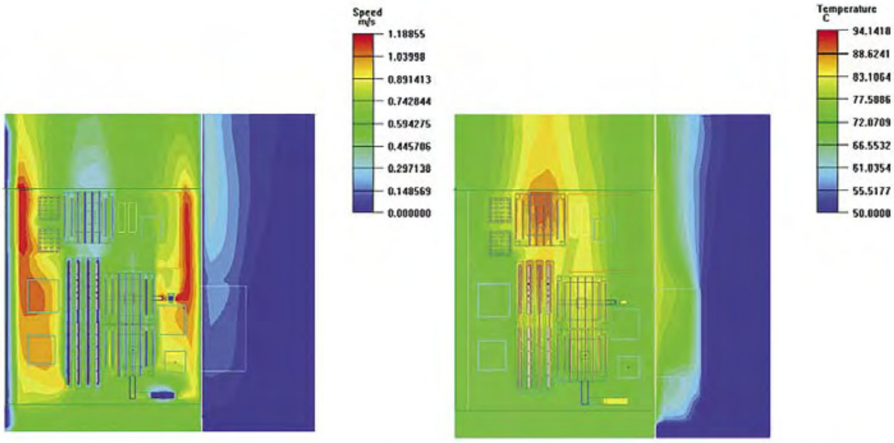


Figure 2

materials can lead to substantially different analysis results compared to testing. Accurate thermal analysis also comes from performing subsequent testing, comparing the two to discover causes of discrepancy, and refining the analysis (or in some cases the testing) for better agreement. The best accuracy comes from iterating this process over many cases, or product designs. This leads to a high level of confidence, allowing rapid *what if* evaluations for continuous innovation.

Commercial air cooling

Air-cooled SBCs tend to be more cost-sensitive than military systems, which means that the more expensive cooling approaches, such as conduction cooling and spray cooling, are not viable options. However, air-cooling poses its own set of challenges. Getting the heat out of high density, high power processors and into the ambient air flow within a specified temperature budget and limited space is no small task.

One method for improving air cooling is to duct more air to desired locations, although the space taken up by the ducts needs to be balanced against any improvement. Reducing the thermal resistance of air-cooled heat sinks is another effective way of creating higher cooling efficiency. One example of this is the use of offset fins to create higher convection coefficients (Figure 3). This results in increased cooling ability within the same volume as a standard plate fin heat sink, with no additional weight.

As discussed earlier, another important approach for solving the cooling problem in air-cooled systems is to conduct detailed airflow analysis of the SBC's component layout. By studying the power dissipation of the components and the airflow over them, the designer can place hot components where they can best take advantage of the airflow so they can be cooled more effectively.

One important decision that effects cooling is the placement of the CPU. The position of the CPU can affect two critical aspects of board design; the signal flows and the heat flow. For example, because VME interface signals are in P1, it can help to place the CPU near P1, which is at the top of the board when engaged vertically in the chassis. This location should also be near the board edge so it does not create a shadow effect on lower profile components downstream from the CPU.

Printed Circuit Boards (PCBs) are currently playing a larger part in cooling than in the past. The main material for PCBs is FR4 Epoxy Glass, which is an industry standard. To help spread heat, thick copper planes are used for power and ground. These planes are also used to prevent signal noise radiation that can cause crosstalk by alternating signal layers and copper plane layers.

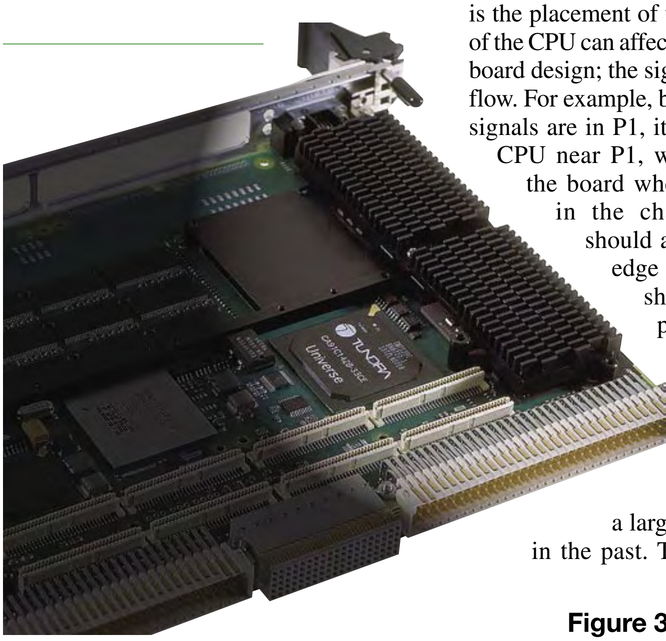


Figure 3

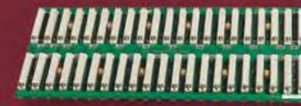
Intelligent use of thermal interface materials is another technique for optimal heat transfer. There is typically a small gap between the CPU case and its heat sink because their surfaces are not smooth. For this reason, thermal designers employ various methods to transfer the heat from the case to the heat sink.

One conventional method uses thermal grease. This material is acceptable in normal temperature ranges with limited reliability requirements. However, with the higher temperatures experienced on today's hot devices and longer reliability requirements, *pump out* of the thermal grease during thermal cycling (for example, on/off cycles) is an issue. In addition, thermal grease is messy to work with.

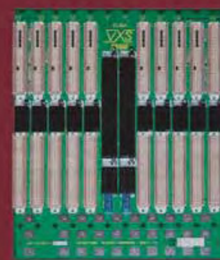
A better alternative is to use thermal tape made from a phase change material with ultra low thermal resistance such as 0.02°C sq.in/W. This material is made of an elastic layer that is tacky at room temperature allowing for easy application. It flows upon the material's first excursion above its phase change temperature, and under compression, it fills the irregularities in both mating surfaces. This reduces thermal resistance in two ways; it decreases the gap thickness, and it eliminates air voids.

Military conduction cooling

For more demanding environments such as military applications, conduction cooling is the preferred approach. A traditional conduction cooling approach using a heat frame, presents challenges stemming from cost, component height limitations, and weight constraints. In use for over ten years, the basic heat frame structure consists of metal bars that run the height of the board, and connect to metal ridges at the board sides. The heat frame has gone from cooling DIP packages at their bottom, to cooling Surface Mount (SMT) packages from the top. They have also gone from cooling 10 W to cooling 100 W. This versatility has allowed conduction cooling to keep up with packaging and cooling changes over the years, and the forecast is for more of the same.


VME


CompactPCI


VXS

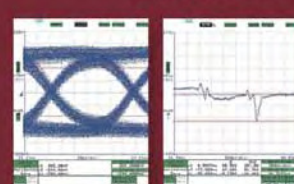

ATCA

Backplanes Geared For Performance

Get on the fast track to success with high-performance backplane design from Bustronic. Our drive is to be the most thorough, knowledgeable, and successful company in backplane technology. We'll get your design done right the first time—saving you time, effort, and money. And our advanced simulation/backplane characterization tools can help us to perfect your complex designs. Come see for yourself why Bustronic is the premier choice for backplanes in the mil/aero industry.

To learn more about the latest technologies in mil/aero backplane design, and for industry news, design tips, and more, visit www.bustronic.com/resource.

Simulation/Backplane Characterization



bus tronic
Elma Bustronic

Tel: 510.490.7388 www.bustronic.com info@bustronic.com

MIL power on the rise

Power levels for military SBCs are continually on the rise. Figure 4 is a plot of the power levels for military 6U modules (160 mm deep, 20.32 mm pitch) with two PMC mezzanine cards by introduction year over the last 12 years. A best-fit curve has been applied to the data (correlation coefficient of 0.9115), and the trend has been extrapolated out five years.

As seen above, there will be requirements to cool over 200 W on a 6U card in the not too distant future – a tall order with today's cooling technologies. Such a high power level may seem unlikely given limitations like board real estate and system-level cooling constraints (such as chassis power dissipation limits). However, electronic components continue a relentless march towards miniaturization and integration, allowing ever-increasing functional (and power) densities. System cooling is being addressed with approaches like liquid cooling, which is discussed later.

In addition to the forecasted high power levels, other trends amplify the cooling challenge, particularly for harsh environment applications such as military and aerospace. The scarce supply of military temperature components has led to the use of parts with lower operating temperatures. The finer geometries used in current and upcoming devices are resulting in higher leakage currents, in addition to increased power density. Finally, the cooling hardware faces ever-increasing competition with electronic hardware for space and weight, two critical factors on military platforms. And in the end, the product must work in all the harsh environments typically encountered by these platforms, such as high temperature, sand and dust, and salt spray environments.

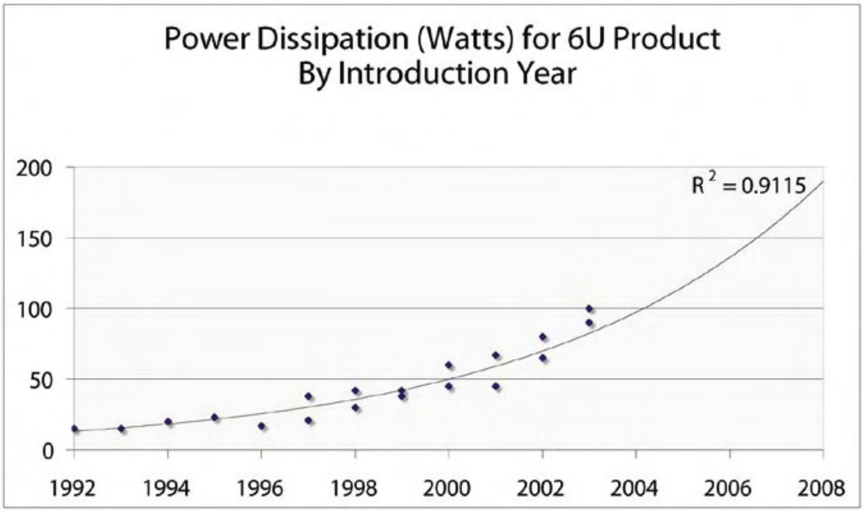


Figure 4

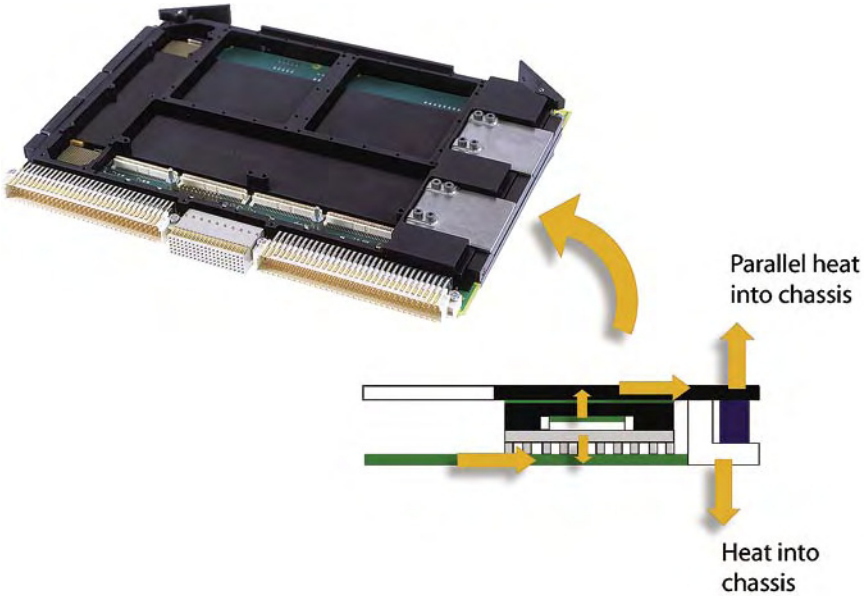


Figure 5

Board-level cooling innovations

The power levels predicted in Figure 4 are challenging the ability of both air-cooling and conduction-cooling approaches to successfully cool products. An example of innovation in conduction-cooling for 6U modules is the use of both sides of the chassis slot for heat transfer (Figure 5).

This substantially increases the amount of heat that can be removed from the module due to an effective doubling of heat transfer area at the card to chassis interface. This interface is a notoriously troublesome location with significant temperature rise due to contact resistance. The approach shown introduces a parallel thermal resistance, which greatly reduces the overall contact resistance.

Putting cooling solutions to the test

Once cooling solutions are conceptualized, designed, and analyzed for performance, the true test comes from testing the SBC at conditions that appropriately simulate harsh environments. The design must be able to cool at high temperatures, and withstand long periods of thermal cycling between hot and cold. It must also be able to withstand high levels of shock, and high vibration levels and durations. Last but not least, it must survive other environments such as humidity, salt spray, sand and dust.

System-level cooling solutions

So far we have focused on cooling at the component and card/module levels. However, electronics heat generation is a system level issue that also needs to be addressed at the chassis and platform level. Liquid cooling at the chassis and platform level is attractive due to the higher densities and specific heats (thermal capacitances) of liquid coolants compared to air. Also, liquids are incompressible, which increases coolant-moving efficiency. The result is smaller heat exchanger components, transport lines, and coolant movers such as pumps. There are, of course, concerns with using liquid cooling, including leakage/spillage and increased cost. However, these issues have been and continue to be addressed on new and existing platforms, and the installed base of liquid cooled electronics continues to expand.

Spray cooling

Spray cooling has also made some inroads into military applications, mostly with direct spray approaches at the chassis level. These approaches spray a mist of coolant directly onto electronic devices, and heat vaporizes the coolant. The vapor is then condensed either within the chassis, or at a remote heat exchanger. Conceptually, spray cooling offers the highest cooling potential of the methods discussed due to the large amount of heat that the liquid to gas phase change can accommodate. However, several obstacles will need to be overcome before it can fully realize this potential, including complexity/reliability, the establishment of a line of sight to hot devices, and maintainability in harsh environments.

Liquid cooled chassis

The use of liquid cooled chassis could greatly increase the longevity of conduction-cooled modules, which have been standardized by specifications such as IEEE 1101.2. As an example, an existing circuit card that dissipates approximately 100 W could dissipate about 150 W (50 percent more power) with a 10°C drop in card edge temperature enabled by liquid cooling the chassis. Of course, the increased temperature drop at the card to chassis interface (due to the increased power) would have to be accounted for.

Liquid Flow Through (LFT) modules

Liquid cooling is also used in Liquid Flow Through (LFT) modules. As the name suggests, these are modules with liquid flowing through them, thereby bringing the coolant closer to the heat generating electronics. This allows for significantly higher power levels on modules. Even with modest flow rates, over 400 W could conceivably be cooled on a 6U module. Alternatively, lower power modules could be cooled to improve component reliability. A custom LFT module developed by Boeing is shown in Figure 6 (image courtesy of Jim Robles of Boeing Corporation).

VITA 46 and 48

LFT modules are currently being standardized in the new industry standards VITA 46 (VPX) and VITA 48 (ERDI). The VITA 46 working group is focusing on the standardization of high-speed connectors on current form factor standards IEEE 1101.1 (for air-cooled modules) and IEEE 1101.2. Rather than burden VITA 46 with new cooling changes, VITA 48 was initiated as a complementary mechanical standard. Also, VITA 46 has an objective of standardizing modules for a two level maintenance option, which requires a change in module thickness to support covers. The two level maintenance option will be addressed in either VITA 46 or 48. Depending on current discussions and the progress of both standards, VITA 46 and 48 may end up merging into one coherent standard.

Conclusions

Power levels on both commercial and military COTS modules continue to rise, stretching the capacity of traditional cooling approaches. Innovative solutions continue to extend the limits of conduction and air-cooling in order to meet future requirements. The highest power applications will eventually demand new approaches such as LFT modules. COTS versions of LFT modules will be standardized in time to enable these applications. Tomorrow's high-power, high-density designs will increasingly require better cooling schemes, making the use of liquid-cooled and spray-cooled techniques an inevitable addition to the use of heat frames. ECD

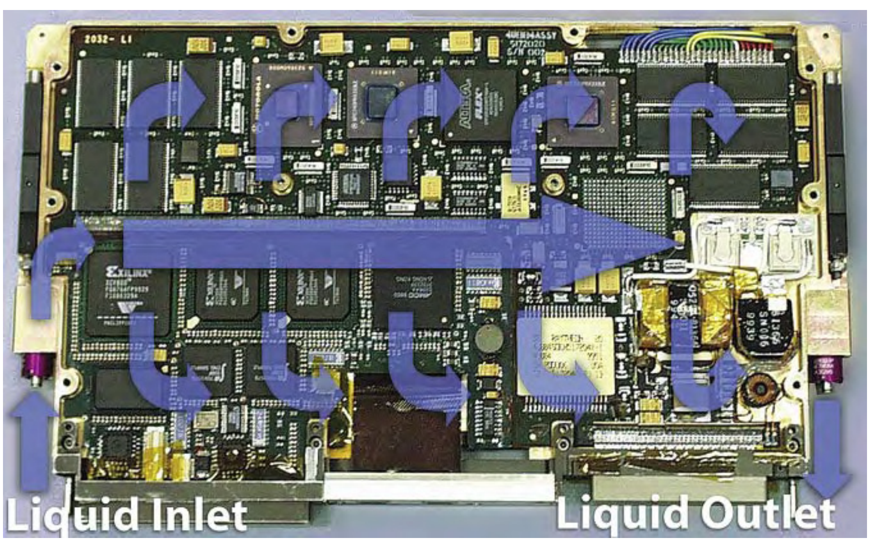


Figure 6

Ivan Straznicky *is a Senior Staff Mechanical Engineer for Curtiss-Wright Controls Embedded Computing (formerly Dy 4 Systems). He has 17 years of manufacturing and mechanical engineering and management experience in the military and aerospace industry. His responsibilities include advanced thermal and packaging technologies for the company's products. Straznicky has a degree in Mechanical Engineering from McGill University in Montreal, Canada.*

Phuc Nguyen *is a Senior Hardware Design Engineer for Curtiss-Wright Controls Embedded Computing (formerly Synergy Microsystems). He has over 20 years of hardware design experience, and helped design the heat frame concept that was used in all conduction-cooled Synergy Microsystems products. Nguyen has a degree in Mechanical Engineering from a university in South Vietnam, and a degree in Digital Microprocessor Design from the University of California, San Diego.*

For more information, contact Ivan or Phuc at:

Curtiss-Wright Controls Embedded Computing

741-D1 Miller Dr. S.E.

Leesburg, VA 20175

E-mail: ivan.straznicky@curtisswright.com

E-mail: phuc@synergymicro.com

Website: www.cwcembedded.com

FPGAs are poised for a change

By Steve Mensor

The debate over whether FPGAs will replace standard cell ASICs has not fundamentally changed over the last 15 years. ASICs are still bigger, faster, and less expensive per device than FPGAs. In fact, the ratio of silicon logic efficiency and core performance between FPGAs and ASICs is still roughly unchanged.

Why is it then, that ASIC design starts are declining and FPGA revenues are growing faster than the overall semiconductor industry? Two reasons: first, the economics of building standard cell ASICs no longer make sense for most applications; and second, FPGAs are big enough and fast enough to address a majority of today's application requirements.

As more advanced process technologies become available, both of these factors will only further accelerate the market shift to programmable logic.

Vendor shifts

This market shift affects semiconductor vendors as well. Many markets that semiconductor companies target are not large enough to justify the suffocating expense to develop new chips on advanced process nodes. In fact, Dataquest predicts that within 10 years, 40 percent of today's semiconductor vendors will no longer be shipping silicon. Many of the companies that leave the semiconductor space will deliver their product as Intellectual Property (IP) through programmable solutions, or they will cease to exist all together.

Value proposition of FPGAs and ASICs

For years, people have argued the benefits of FPGAs versus those of ASICs. FPGAs offer the benefits of fast time-to-market and low development costs, while ASICs have the advantage of lower device costs, higher logic density, and higher performance.

However, this is not the whole story. FPGA design verification time is significantly shorter than that for ASICs because FPGA designs can be tested in hardware. Additionally, FPGAs allow companies to upgrade their products that are deployed in the field by remotely configuring the FPGA, saving time and money.

Conversely, the ASIC design and verification process is significantly more demanding, as design errors on ASICs require expensive and time-consuming ASIC re-spins. From a total-cost-of-implementation perspective, FPGAs make sense for low- to moderate-volume applications. ASICs make more sense for high-volume applications where the lower ASIC device price outweighs the higher ASIC development costs.

Silicon efficiency of FPGAs versus ASICs

As a review, the FPGA's basic building block is the logic element. A logic element has a Look-Up Table (LUT) that emulates any function up to four inputs plus a register and some other specialized circuitry. Benchmarks show that a logic element typically implements about 12 ASIC gates of logic.

ASICs have higher silicon efficiency than FPGAs for implementing logic because ASICs do not have the overhead of programmable circuitry. In the mid-1990s, leading ASIC vendors claimed logic capacity of 7,000 to 10,000 gates per square millimeter on 0.5 micron processes. Using the 12 gates of logic per logic element ratio, FPGAs at that time had a die-size efficiency of about 500 gates per square millimeter. This 15x to 20x ratio of greater silicon logic efficiency for ASICs is consistent today.

Today, however, this ratio does not represent the entire story. FPGAs also have embedded functions that are nearly as die-size efficient as the same functions in ASICs. The best example is embedded memory. Currently, the largest FPGAs have close to 10 megabits of embedded memory. The ASIC die-size advantage diminishes when comparing designs that use the embedded functions in FPGAs.

Despite this, ASICs still have a die-size advantage over FPGAs that equates to a cost advantage, but only when the comparison is made for devices built on the same process. Due to the rising cost of developing chips, fewer ASICs will be built on advanced processes. However, FPGAs will continue to be built on the most advanced processes available. Eventually, this process technology gap will eliminate the advantage that ASICs have had over FPGAs.

Changes due to rising costs

In the 1990s, the rising costs of building fabs changed the semiconductor industry. Building a new fab in the 1980s cost tens of millions of dollars, but that grew to hundreds of millions of dollars in the 1990s. As a result, the vertically integrated semiconductor model failed, and the fabless model was created. Even semiconductor giants like Texas Instruments are transitioning to a fabless model for select products. Today, all but a few of the 550 semiconductor companies are fabless.

A similar situation is happening now at the silicon level. Chip development costs continue to rise exponentially, but, ironically, this increase is not due to the higher cost of new fab development. Much has been made of the rising wafer costs for advanced processes, however, the majority of the cost increase in developing chips results from the increasing engineering

FPGAs are poised for a change

costs of design and verification. While the estimated costs to build chips vary, Gerry Worchel, senior analyst at In-Stat/MDR, estimates that the cost of getting chips from design to production is between $40 million and $50 million for a 90 nm product.

Consider the demand needed to justify a $40 million project. There are very few products that have enough end-market revenue to absorb this high development cost. With research and development typically taking 15 to 20 percent of revenue, a custom chip only makes sense for end products that generate revenues in excess of $200 million. Now consider market share. In most industries, no single company controls more than 20 percent market share. This means the target market needs to be at least $1 billion. There are markets that exceed $1 billion in total sales, but not as many as one might think. Cell phones, computers, printers, and MP3 players are good examples of applications where custom chips make sense. However, even companies that sell into these markets will face this issue in the future as product development costs continue to rise.

When companies can no longer afford to build custom chips, they must look for economic alternatives. The only alternative that makes sense is programmable chips.

Programmable chips include programmable logic, processors, and memories. All of these devices can be configured to perform unique functions based on the requirement of the targeted application. In the future, it is likely that typical systems will contain only programmable chips, analog devices, and passive components.

FPGA process advantage

FPGAs are poised for rapid growth for most high-volume applications. This may not seem intuitive because people perceive FPGAs as expensive. While some FPGAs do sell for over $1,000, the majority of FPGAs sell for under $50, and in many cases, for as low as a few dollars. This wide price range is primarily because FPGAs are SRAM-based structures scaled to create different density devices.

Die sizes range from below 20 sq mm, where the chip is typically pad-limited, to beyond 500 sq mm, or close to the size of a postage stamp. The die-size range for FPGAs has been consistent for years, but by moving to more advanced processes, FPGAs gain more capacity and higher performance. While it is becoming cost prohibitive for companies to build custom solutions on advanced process nodes, FPGAs will continue to follow Moore's law down the process curve.

From a foundry point of view, FPGAs are ideal devices for bringing up new process nodes. Being basically memory structures, they are standard arrays that are optimal for defect identification, and the bigger FPGAs with large die sizes are useful for reducing the process defect density. FPGAs have dense metal arrays for routing, which is optimal for identifying and reducing metal defects. They also tend to use the highest performance transistors, which helps identify and reduce transistor defects.

Future of programmable logic

To understand where FPGAs are going, it is necessary to understand where they came from. Originally, FPGAs were an array of a couple hundred logic elements used for building simple glue-logic functions. Today, FPGAs have close to 200K logic elements, embedded memory, dedicated Digital Signal Processing (DSP) blocks, Phase-Locked Loops (PLLs), dedicated external memory interfaces, and I/O pins that support multiple single-ended standards, differential standards, and clock-data-recovery transceivers.

FPGAs have come a long way since they were introduced in the mid 1980s. A good benchmark of this progress is evident with the Altera Nios II embedded soft processor. The Nios II embedded soft processor is a configurable processor where one or more instances can be programmed into any of Altera's mainstream or advanced FPGAs. Nios II processors have several configurations depending on whether the user is more concerned about speed or logic efficiency. It requires about 1,500 logic elements for the processor and the supporting subsystem which includes timers, UART, multiple general purpose I/O ports, SDRAM Memory Controller, and interfaces to external flash, SRAM, an Ethernet MAC/PHY, and LCD display controller.

FPGA development

The most cost sensitive devices in FPGA families are mid-density devices, and in 1995, a 0.5 micron process FPGA mid-density device included around 2,500 logic elements at a cost of $150 per 1,000 units.

If the Nios II processor had been available in 1995, it would have consumed 60 percent of the 2500 logic elements in the mid-density device, and would therefore have had an effective cost of $90 per 1,000 units.

In 2000, the most advanced FPGAs were built on 0.18 micron processes, and the effective price of the Nios II processor benchmark was $35 per 1,000 units.

Today in 2004, using a low-cost FPGA built on a 130 nm process, the effective price of the Nios II processor benchmark is only $3 for 1,000 units.

In 2009, leading-edge FPGAs will be built on 45 nm processes, and will have over 500K logic elements. A Nios II embedded soft processor on these devices will use less than 1 percent of the device resources, and cost well below $1 for 1,000 units.

Today, FPGAs are the lead driver for advanced processes and have a large customer base that allows FPGA vendors to absorb the ever-increasing cost of building semiconductors on advanced process nodes. Limited by market size, vendors building custom chips can no longer benefit from Moore's Law by migrating their products to more advanced process nodes. For similar reasons, standard cell ASIC vendors are charging much higher Non-Recurring Engineering (NRE) prices for products on advanced process nodes. As a result, the technology gap between FPGAs and ASICs will narrow dramatically. In the near future, FPGAs will offer similar performance, density, and cost structures as ASIC alternatives without the exorbitant development costs. ECD

Steve Mensor *is the Senior Director of Product Marketing for New Products for Altera Corporation, and has been with the firm since 1988. He has held management positions in the Customer Applications, Customer Marketing, and Product Marketing organizations. He is responsible for marketing of Altera's high-density Programmable Logic Devices (PLDs). Mensor holds a BSEE degree from the University of California, Santa Barbara and an MBA from Santa Clara University.*

For more information, contact Steve at:

Altera Corporation
101 Innovation Drive
San Jose, CA 95132
Tel: 408-544-7156 • Fax: 408-544-6425
E-mail: smensor@altera.com
Website: www.altera.com

Plug SBE's TCP/IP Offload Engine (TOE) into your system to see the performance boost for yourself...

All TOEs should process TCP/IP at network speeds, provide full segmentation and assembly, terminate multiple simultaneous sessions, and minimize transaction latency, without host intervention. However, not all TOEs are alike...depending on the individual manufacturer's target market, the devices vary in their ability to fully handle these essential criteria.

Adding a TOE to your existing system is only cost-effective if it can truly heighten performance at a fraction of the cost of purchasing an additional server. Let results speak for themselves. Only one TOE board has proven to provide peak performance across all four metrics of TOE effectiveness. While other TOE vendors are capable of satisfying one, two or maybe three of the critical TOE performance metrics, SBE is today's only source to deliver Gigabit Ethernet throughput at line rate, over 70% reduction in CPU utilization, 32 microsecond transaction latency, and support for high session count applications, all on one board.

See the results for yourself...

Contact us to qualify for a free test drive of the SBE TOE board. Plus, ask about how to win an iPOD® mini.*

Phone: 800-214-4723
Email: info@sbei.com
Web: www.sbei.net/tryTOE.htm

SBE®

Linux *On Demand*

flexibility on demand | 925-355-2000 | info@sbei.com | www.sbei.com

DEVICES

Company name	ARM	Bridging	Bus interface	Component-level	DSP	FPGA	IP	MIPS	Pentium	PowerPC	SoC	Other	Website
Acromag						•							www.acromag.com
Actel						•							www.actel.com
AeroComm				•									www.aerocomm.com
Alpha Data						•							www.alpha-data.com
Alphi Technology						•							www.alphitech.com
Altera Corporation						•							www.altera.com
AMCC										•			www.amcc.com
Ampro Computers						•							www.ampro.com
AMREL												•	www.amrel.com
Anadigm						•							www.anadigm.com
Analog Devices	•			•	•								www.analog.com/processors
APS						•							www.associatedpro.com
Arista									•				www.aristaipc.com
Atmel	•		•	•		•						•	www.atmel.com
austriamicrosystems				•									www.austriamicrosystems.com
AVX Corporation				•									www.avxcorp.com
BittWare						•							www.bittware.com
Broadcom								•					www.broadcom.com
C&D Technologies				•									www.cdpoweronline.com
Cambridge Consultants												•	www.cambridgeconsultants.com
Cavium Networks								•					www.cavium.com
CoEv Magnetics				•									www.circuitprotection.com/magnetics.asp
Comsys							•						www.comsysmobile.com
Comtech AHA Corporation												•	www.aha.com
Connect One Semiconductors	•												www.connectone.com
CPU Technology												•	www.cputech.com
Curtiss-Wright Controls						•							www.cwcontrols.com
Dallas Semiconductor												•	www.dalsemi.com
DNA Enterprises						•							www.dna-cs.com
ELMA Electronic				•									www.elma.com
Eonic B.V.					•								www.eonic.com
Eureka Technolgoy, Inc.		•	•										www.eurekatech.com
Fairchild Semiconductor				•									www.fairchildsemi.com
Fox Electronics				•									www.foxonline.com
Grid Connect												•	www.gridenabled.com
Hunt Engineering						•							www.hunteng.co.uk
Infineon												•	www.infineon.com
Inicore							•						www.inicore.com
Intel									•				www.intel.com
Interactive Circuits & Sys.						•							www.ics-ltd.com
Jacyl						•							www.jacyltechnology.com

Company name	ARM	Bridging	Bus interface	Component-level	DSP	FPGA	IP	MIPS	Pentium	PowerPC	SoC	Other	Website
Linear Technology				•									www.linear-tech.com
LSI Logic					•								www.lsilogic.com
Mentor Graphics						•							www.mentor.com
Mercury Computer Systems						•							www.mc.com
Micrel				•									www.micrel.com
Micronas Semiconductor				•									www.micronas.com
MIPS Technologies								•			•		www.mips.com
MOSCHIP				•									www.moschip.com
M-Tron Industries				•									www.mtron.com
Nallatech						•							www.nallatech.com
ON DEMAND							•						www.ondemand.co.at
ON Semiconductor				•									www.onsemi.com
Pentek						•							www.pentek.com
Pericom		•		•									www.pericom.com
Picor				•									www.picorpower.com
PLDApplications			•										www.plda.com
PLX Technology		•											www.plxtech.com
Protocol Tools			•										www.protocoltools.com
Pulse				•									www.pulseeng.com
QualCore Logic							•						www.qualcorelogic.com
QuickLogic								•					www.quicklogic.com
Rabbit Semiconductor												•	www.rabbitsemiconductor.com
RF Engines Limited							•						www.rfengines.com
RF Microdevices											•		www.rfmd.com
Saelig				•									www.saelig.com
SaRonix				•									www.saronix.com
SBS Technologies						•							www.sbs.com
SMSC									•				www.smsc.com
Spectrum Sig. Proc.						•							www.spectrumsignal.com
Summit Microelectronics				•									www.summitmicro.com
TDK Semiconductor											•		www.tdksemiconductor.com
Tews Technologies						•							www.tews.com
Texas Instruments					•								www.ti.com
Transdimension		•	•										www.transdimension.com
Tundra Semiconductor		•	•										www.tundra.com
Tyco Electronics				•									www.tycoelectronics.com
Vishay				•									www.vishay.com
Vitesse Semiconductor				•									www.vitesse.com
White Electronic Designs										•			www.whiteedc.com
Xemics				•									www.xemics.com
ZiLOG												•	www.zilog.com

New PRODUCTS

By Eli Shapiro

TABLE OF CONTENTS

BACKPLANE ACCESSORIES

Kaparel

Website: www.kaparel.com
Model: 6U cPCI Backplane Stiffener **RSC No:** 19252
A backplane stiffener that reduces bowing in lower width, 6U backplanes • Reduces

RSC #19252

backplane flex by as much as 70 percent (6U x 3-slot) at a backplane thickness of 3.5 mm/0.137 inch • Can be retrofitted to all Kaparel CPCI RP/RS backplanes without physical changes to the backplane itself

CARRIER BOARD: M-MODULE

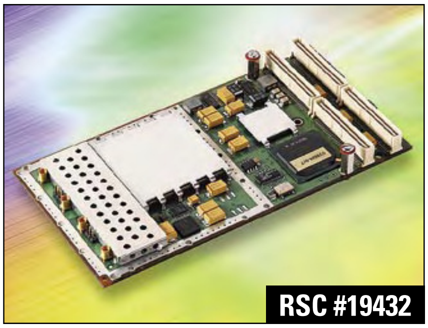
Racal Instruments

Website: www.racalinst.com
Model: EM405D **RSC No:** 18978
Wired and wireless M-Module carriers • Allows creation of small test and data acquisition subsystems, which are accessible via Ethernet or wireless interconnections • Manufactured by C&H Technologies • Supports two ANSI/VITA 12-1996-compliant M-Modules • Available with or without an enclosure • Optional AC adapter • Trigger in-and-out capabilities are available via the D-sub connector

CARRIER BOARD: PMC

Radstone Technology

Website: www.radstone.co.uk
Model: PMC571 **RSC No:** 19432
Single channel software radio rugged mezzanine • Sample rates up to 80 MHz • Processing bandwidth 40 MHz • 14-bit Data Conversion on ADC and DAC functions • Full 64-bit/66 MHz PCI Interface • 64-bit general purpose I/O • Xilinx XC2V4000 Virtex II User FPGA • Air- and conduction-cooled ruggedization levels • Extensive application and Technical Support Data available



RSC #19432

CHIPS & CORES: PowerPC

AMCC

Website: www.amcc.com
Model: PowerPC 440SPe **RSC No:** 19004
An embedded PowerPC processor core • Frequencies from 533 to 667 MHz, with 1,334 DMIPS at 667 MHz • 32 KB instruction and 32 KB data caches • 256 KB L2 SRAM with parity protection • 32/64-bit DDR I/DDR II SDRAM controller for DDR166/333 and DDR333/667 operation • Three PCI Express interfaces, one with eight lanes and configurable as x8, x4, x1, and two with four lanes and configurable as x4 and x1 • 64-bit PCI-X version 2.0 3.3 V interface, up to 133 MHz/DDR 266 • Opaque PCI Express-to-PCI Express and PCI Express-to-PCI-X bridge functionality • Ability to boot from the primary PCI-X bus memory • XOR core accelerates parity generation and check functions • Intelligent messaging unit (I2O specification) • Two-channel DMA included with I2O, one-channel DMA included with XOR • External Bus Controller (EBC) interface (up to 83 MHz) supporting up to three ROM, RAM, or EPROM peripheral devices • One 10/100/1000Base-T Ethernet MAC, full-duplex GMII/MII interface with DMA capability via Memory Access Layer (MAL) • Three serial ports (16750 compliant UARTs) with 64-byte FIFOs • Two IIC controllers • Up to 32 General-Purpose I/Os (GPIO) • General-purpose timers • Universal programmable interrupt controller supports 16 external interrupt sources and 101 internal sources • JTAG, RISCTrace, Chip Onboard Logic Analyzer (COLA), and RISCWatch support

RSC #19004

COMPONENT-LEVEL MODULES

Linear Technology

Website: www.linear-tech.com
Model: LTC4441 **RSC No:** 19515
A 6A MOSFET gate driver • 6A peak output current • Wide input supply range from 5V to 25V • Adjustable Gate Drive Voltage from 5V to 8V • Logic input can be driven below ground • Adjustable blanking time reduces ringing • 30 ns propagation delay • Rated for operation from –40°C to +85°C

Vitesse Semiconductor

Website: www.vitesse.com
Model: VSC060 **RSC No:** 18993
An enhanced, high-density I2C backplane controller • Up to 96 bits of 5 V tolerant, user-definable, bidirectional, general-purpose I/O • Slave mode I2C serial interface • Integrated port bypass, clock recovery and signal detect support for up to 16 drive ports • Eight programmable fan speed monitoring inputs • Eight programmable pulse width modulated fan control outputs • Pairing of GPIO pins for direct I/O signal routing/buffering • Pin-programmable addressing for up to 16 devices on a single serial bus • 100-pin PQFP package • Two clock input ranges: 8.0 to 12.5 MHz (crystal

or external clock source) or 32.0 to 75.0 MHz (external clock source)

RSC #18993

CONNECTOR: OTHER

Amphenol

Website: www.amphenol.com
Model: Ampe-Power Series **RSC No:** 18984
Three families of connectors with RADSOK


RSC #18984

technology, a terminal that features a stamped hyperbolic grid, providing low insertion force, high amperage, high reliability, and a longer cycle life • Families include the P-Lok connector for high-amperage applications, Amphe-Power GT reverse bayonet coupling connectors, and the Amphe-Power AC series of threaded cylindrical connectors • Supports from 50 A to over 1,000 A of continuous duty

DATA ACQUISITION

Measurement Computing

Website: www.measurementcomputing.com
Model: PCI-DAS6031 **RSC No:** 18961
A PCI form factor data acquisition card with identical specifications to the NI PCI-6031E • Provides 64 single-ended/32 differential channels of 16-bit (one part in 65536) analog input • Measurements on any of 14 input ranges can be taken at rates up to 100 KSps • Dual 16-bit analog outputs with 100 KSps per channel update rates • Eight digital bidirectional I/O lines can control relays and solenoids, and can monitor switch and contact closures • Two 16-bit counter/timers • Fully supported by LabVIEW, SoftWIRE, Agilent VEE, and MATLAB • Supports Measurement Computing's Universal Library (UL) for programming in Visual Basic, C#, C++, and other popular languages

United Electronic Industries

Website: www.ueidaq.com
Model: PD2-MF-3M/12 **RSC No:** 18996
A family of multifunction cards with aggregate throughput rates of 3 MHz on either 16 or 64 analog input channels with 12-bit resolution • Available for the PCI bus • Analog front end provides either 16 or 64 multiplexed inputs that feed a 3 MHz, 12-bit A/D converter • Instrumentation amplifiers for either high-level or low-level signals • Standard onboard FIFO is 16 Ksamples, with options for 32 or 64 Ksamples • Dual 12-bit analog outputs at 200 KSps • 16 digital inputs • 16 digital outputs • Three user counter/timers • Onboard DSP runs a dedicated kernel • Software support through the PowerDAQ software suite

RSC #18996

DATACOM: SECURITY

TeamF1, Inc

Website: www.teamf1.com
Model: SSHield **RSC No:** 19612
A robust, standards-based, small-footprint Secure Shell (IETF SECSH) implementation for VxWorks 5.x and AE • Provides SSH protocol client and server support with both SSHv1 and SSHv2 • Includes sftp client and server as well

as scp with flexible library-style APIs • Supports password authentication in addition to public-key user authentication • X.509 certificate support for authentication • Support for Kerberos authentication • Supports custom authentication mechanisms • Modular crypto to scale out unneeded ciphers and hashes • APIs for target-based key generation • Data compression support • Port forwarding for legacy applications and X11 forwarding • Abstracted file I/O system • Works with standard SecureShell client implementations on other platforms • Supports CPU types of either endian-ness, including PowerPC, MIPS, x86, ARM/XScale

ENCLOSURE + CARD RACK + POWER SUPPLY

Kaparel

Website: www.kaparel.com
Model: 3U cPCI Rack-Mounted **RSC No:** 19253
Aluminum vented system, 358 mm deep, for installation in 19" (486.2 mm) rack, cabinets or cases • Accepts PICMG 2.0/2.1 CompactPCI 3.5U x5-slot backplane • Prepared to accommodate CompactPCI boards and drives • Fully wired and tested • Front to back cooling • Front/rear interface for CompactPCI defined injector/extractor handles • Complies with IEC 60297-3/4/5-1xx: IEEE 1101.1/.10/.11

Kaparel

Website: www.kaparel.com
Model: 9U cPCI Rack-Mounted **RSC No:** 19254
9U, 84HP x 300 mm deep, 19" aluminum subrack; accepts up to 14 6U, 32/64-bit CompactPCI cards and up to two double-wide (8HP) 6U-power supplies; two high performance output RiCool Blowers; power switch, and power indicator LED; one PS1241 power backplane • 16HP wide backplane accepts two PS335X 350 W AC or DC power supplies each; two PS1420 CompactPCI backplanes or two PS4400 H.110 CompactPCI backplanes; 14 CompactPCI slots • PS1130 32/64-bit PCI-to-PCI rear bridge card; cable assemblies and wiring harnesses; compatible with the IEEE1101.11 Specification with 80 mm transition modules; provides 13 peripheral slots and one host slot

INTEGRATED DEVELOPMENT ENVIRONMENT

OSE Systems

Website: www.ose.com
Model: OSECK **RSC No:** 19002
An enhanced development environment suitable for wireless, radar, sonar, and media gateway applications • Based on the full-featured OSE message-based real-time operating system architecture designed for distributed, advanced multiprocessor, multicore heterogeneous applications • Supports TI TMS320C64x family, TMS320C54x, C55x, C62x, and C67x, the Motorola StarCore 810x series, Analog Devices TigerSHARC, Agere Systems DSP 16K, STMicroelectronics ST120, and LSI Logic ZSP400

MASS STORAGE: SOLID STATE DISK

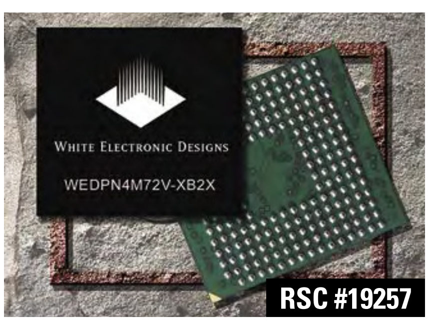
Adtron

Website: www.adtron.com
Model: A25FB **RSC No:** 18951
A Serial ATA solid-state Flash disk • Capacities up to 60 GB in a 2.5" form factor • Sustained read/write rates up to 40 MBps • Adtron secure erase technology • Hot plug features to reduce downtime • Standard 7-pin signal connector with a 15-pin power connector

MEMORY: GENERAL PURPOSE

White Electronic Designs

Website: www.whiteedc.com
Model: WEDPN4M72V-Xb2X **RSC No:** 18957
A 4M x 72 synchronous DRAM module • Packaged in a 21 mm x 21 mm, 219 plastic ball grid array • Weighs 2 grams (typical) • COTS compliant • Upgradeable to 8M x 72 density within the same footprint • Uses an internal pipelined architecture • Frequency ranges of 100 and 125 MHz • All signals registered on the positive edge on system cycle clock



POWER SUPPLY

Absopulse Electronics

Website: www.absopulse.com
Model: FC 500-C Series **RSC No:** 18989

A 500 VA AC/AC frequency converter with a sine wave output • Compact size and light weight • Sinusoidal wave shape • 500 VA of output power • Full electronic protection • Field-proven design topology • Designed to meet MIL-461 • Cooled by a built-in fan • Extended operating temperature of –40°C to +65°C

Cherokee International

Website: www.cherokeellc.com
Model: CAR1248 **RSC No:** 18977

A front end/rectifier for low-profile, distributed power architecture applications • 1U high footprint; provides power density of 19 W per cubic inch • Delivers a 48 V single output at 25 A, along with a 5 V standby voltage • Integrated redundant diodes and a thermally controlled fan, enabling –25°C to +70°C operation and –40°C startup capability • Hot-swap and fault redundancy, active single-wire current sharing, output voltage programming, current readout, remote on/off, and front-plate LEDs • Protection features include input over- and undervoltage, output overvoltage, overtemperature, and overcurrent

PROCESSOR: PENTIUM III

Trenton Technology

Website: www.trentontechnology.com
Model: SLE **RSC No:** 19167

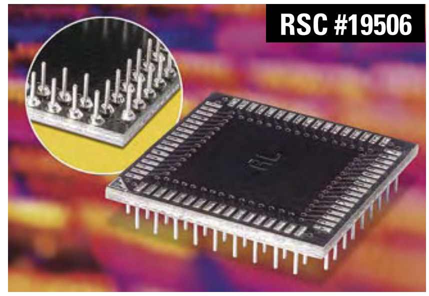
A single board computer in PICMG 1.0 PCI form factor • Dual Intel Pentium III processors at 733 MHz to 1.26 GHz • ServerWorks ServerSet III LE chipset with a 133 MHz FSB • Ultra3 SCSI interface with a QLogic ISP10160A SCSI controller • Dual 10/100Base-T Ethernet interfaces • Dual Ultra DMA/33 interfaces • Up to 2 GB of 133 MHz SDRAM • SMP for multi-threaded applications • Intel 69030 SVGA video interface with 4 MB of on-chip memory

PRODUCTION TOOLS

Aries Electronics

Website: www.arieselec.com
Model: Correct-A-Chip **RSC No:** 19506

An adapter that enables the use of Atmel ATV5000 and Cypress CY7C342B, 68-pin PLCC packages with boards designed for the obsolete pin grid array packages, without board redesign or rework • 68 PGA pins on the bottom and either solder pads (part number 68-7447-10) or a 68-pin PLCC socket (68-7747-12) on the top • Limited-depth pins are stacked into the body and soldered • Male pins are brass alloy 360 with 200 µ tin over 100 µ minimum nickel per QQ-N-290 • Socket pins are phosphor bronze alloy per



QQ-V-50 • Adapter body is 0.062" thick glass-filled FR-406 high-performance epoxy laminate with a glass transition temperature of 170°C • Body has one ounce copper traces and is UL94V-0 rated

PROTOTYPING AND DEBUGGING AIDS

Hitex Development Tools

Website: www.hitex.com
Model: TC1130 Tool Chain **RSC No:** 18975
A professional tool chain for the 32-bit TC1130 microcontroller in the TriCore family by Infineon • Complete support for architecture-specific functionalities • HiTOP5 debugger used as the GUI • HiTOP5 permits fast programming of Flash memories and offers convenient display and control of the special function registers


RSC #18975

SOFTWARE: DEVELOPMENT TOOL

MathWorks

Website: www.mathworks.com
Model: RF Tools **RSC No:** 18983
RF design and simulation tools (RF Blockset and RF Toolbox) for MATLAB and Simulink • Designed to expand the scope of model-based design for signal processing and communications engineering applications • Provides features and benefits for RF system design, analysis, development, and implementation • RF Blockset works within the Simulink environment and offers a library of blocks to model the behavior of RF amplifiers, mixers, filters, and transmission lines • RF Toolbox extends the MATLAB environment by providing pre-built design and analysis functions and a graphical tool for working with, analyzing, and visualizing the behavior of RF components • Available for Windows, UNIX/Linux, and Macintosh platforms

SOFTWARE: NETWORKING

LVL7

Website: www.LVL7.com
Model: FASTPATH **RSC No:** 18998
Distributed networking software designed for equipment vendors developing enterprise and service provider products, control backplane solutions, industrial automation products, or hybrid multifunction switching devices • Fully integrated software allows OEMs, TEMs, ODMs, and CMs building 10/100, Gigabit, or 10 Gigabit Ethernet solutions to develop production-ready products in less than 90 days • Available in the FASTPATH 2000 and FASTPATH 4000 platforms

SOFTWARE: OPERATING SYSTEM

Jungo

Website: www.jungo.com
Model: GO-HotSwap V 6.22 **RSC No:** 18869
Enables Hot Swap of CompactPCI peripheral devices under operating systems that do not natively support hot swap • Multi-operating system support • Cross operating system capabilities • Configuration Manager: Enables hot swap with legacy PCI drivers • Hot swap driver development toolkit, including graphical hardware debugger • Free full featured 30-day evaluation version hot swapping legacy PCI devices) • Enables dynamic loading and unloading of legacy PCI drivers to enable hot swap with non hot swappable drivers • Pre-configured according to user's definitions to run any application or script upon detection of a hot swap event

TELEPHONY: VoIP

Octasic

Website: www.octasic.com
Model: OCT8304 **RSC No:** 18981
A 1023-channel, full-duplex packetization engine for Voice Over IP and Voice Over AAL2 applications • 1000/4000+ voice packetization/aggregation capacity • Less than 250 μs of latency • Less than 2.5 W of device power • Simultaneous support of IPv4, IPv6, and AAL2 • Full AAL5 support including: Classical IP, LANE (V1/V2), MPOA • Supports MII (Ethernet), UTOPIA, and POS-PHY network interfaces •

Two network ports • 4096 timeslot H.100/H.110 compliant TDM interface • Supports BLES • Dynamic voice trunking • Direct interworking with Octasic high-density ADPCM (OCT7102) and echo cancellation (OCT6100 Series) devices • Silence suppression • Read and write cache for fast accesses • 3.3 V and 2.5 V power supply • Packaged in a 608 EPBGA

TURNKEY SYSTEM

AAEON Electronics

Website: www.aaeon.com
Model: MPC-8890 **RSC No:** 18956

A media PC for POS/kiosk applications • Supports low-voltage Pentium M and Celeron M processors at up to 2.0 GHz • Supports 4/5/8 wire resistive touch panels via an onboard touchscreen controller • Dual view function • 5.1 channel audio output • Four digital I/O • Four COM and USB 2.0 ports • Integral mini-PCI slot

MPL

Website: www.mpl.ch
Model: PANEL-PIP **RSC No:** 18954

A range of IP65/NEMA4-protected panel PCs • No fans or air vents • Based on the Packaged Industrial PC (PIP) family from MPL • Available in a stainless steel or aluminum housing • Display sizes from 12.1" to 19" • Available with a wide range of processors • 3D graphics chip with dual display support • Internally expandable via PC/104 and PC/104-*Plus*

VIDEO: FRAME GRABBER

Sensoray

Website: www.sensoray.com
Model: 617-4 **RSC No:** 18972

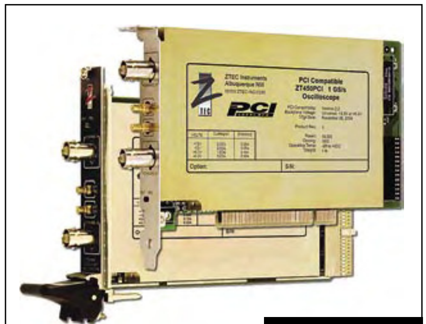
A traffic monitoring frame grabber • Combines the functions of a frame grabber, hardware JPEG compression, and digital I/O on one PCI board • Accepts four composite video inputs and routes them to separate video decoders for digitizing • Use of four decoders eliminates losing frames during channel switching • Capture rate of 30 fps at 640 x 480 resolution (25 for PAL), or 120 fps at 320 x 240 (100 for PAL) • Sensoray's software supports multiples 617-4s in the same PCI bus • Four video inputs are routed to four output channels for display on video monitors • Caption buffer allows overlaying text on each frame • Supplied with a software developer's kit that includes drivers for Linux and Windows 98/NT/2000/XP

WAVEFORM DIGITIZERS/DIGITAL OSCILLOSCOPES

ZTEC

Website: www.ztec-inc.com
Model: ZT450 Family **RSC No:** 19510

A family of 8-bit, high-speed, low-power Digital-Storage Oscilloscopes (DSO) with benchtop instrument features • Features include flexible signal conditioning, advanced triggering, average and envelope acquisition modes, and onboard signal processing • Available with two (PCI and CompactPCI/PXI) or four (VXI) channels • Provides simultaneous sampling on all channels at 1 GSps, or interleaved sampling on half of the channels at 2 GSps • With repetitive-time sampling, the DSO can sample up to 50 GSps • 500 MHz bandwidth on all input settings with up to 32 MB per channel waveform memory • Available with a 1 GHz analog bandwidth option • Auto setup, auto calibration, setup configuration save and recall, reference waveforms, gated measurements, mask testing, 50 and 1 MΩ input impedance, input protection, and probe attenuation • Built on the ZTEC Universal Instrument Platform (UIP) • Comes with plug-and-play instrument drivers, which can be used with software development environments such as LabVIEW, LabWindows/CVI, Visual Basic, and C/C++



RSC #19510

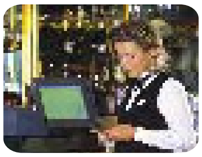
▶ Maximum User's Image Experience

- Dual Independent Display
- Larger Display Like Plasma
- Dazzling Graphics And Audio
- Video on Demand
- Digital Broadcasting

▶ Minimum Waiting Time, Seamless Interaction

- Intel® Pentium® 4, Hyper Threading
- Intel® Low Power Pentium® M/Celeron® M
- Intel® Extreme Graphics 2, DDR SDRAM
- High Speed I/O, USB 2.0, POWER USB, IEEE1394
- Wire and Wireless Ethernet Connection

◂◂ ATM/Kiosk



◂◂ POS

◂◂ Gaming

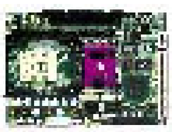
◂◂ Medical

▶▶ Portwell Solution

PEB-3715VLA

5.25" ESB

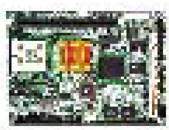
5.25" ESB based on Intel® Pentium® 4 or Celeron® processor with DDR, AGP 4X VGA/Panel, Dual Display, Fast Ethernet and Audio



PEB-3730VLA

5.25" ESB

5.25" ESB based on Intel® Pentium® M or Celeron® M Processor with DDR SDRAM, AGP 4X, VGA/Panel, Dual Display, LAN and Audio

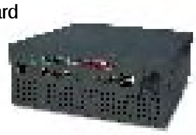


AREMO-7013

19" 2U industrial rack-mount chassis for embedded M/B

PEC-5100

Chassis for 5.25" embedded board



RSC #60 @ www.embedded-computing.com/rsc

Intel® Communications Alliance Associate Member SILVER

Microsoft® Windows® Embedded Partner

One Step Ahead in Communications!

American Portwell Technology, Inc.
6851 Mowry Avenue Newark, CA 94560 USA
Tel: +1-510-790-9192 Fax: +1-510-790-9191
E-mail: info@mail.portwell.com
http://www.portwell.com/embedded/